



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07253949 A**(43) Date of publication of application: **03.10.95**

(51) Int. Cl. **G06F 15/16**
G06F 7/72

(21) Application number: **05326008**(22) Date of filing: **30.11.93**

(30) Priority: **30.11.92 IL 92 103921**
16.02.93 IL 93 104753
06.09.93 IL 93 106923

(71) Applicant: **FORTRESS U & T LTD**

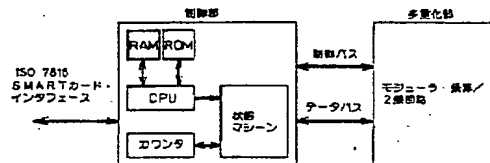
(72) Inventor: **CRESSEL CARMI D**
HENDEL DAVID
DROR ITAI
HADAD ISAAC
ARAZI BENJAMIN

(54) **MICRO ELECTRONIC DEVICE AND METHOD FOR EXECUTING MODULAR MULTIPLICATION AND MODULAR POWER** COPYRIGHT: (C)1995,JPO

(57) Abstract:

PURPOSE: To reduce the time required for executing modular multiplication, etc., based on the Montgomery method with respect to a micro electronic device and method for executing modular multiplication and power to large numbers.

CONSTITUTION: A micro electronic device is constituted of fractionized and switching-controllable compact synchronous micro electronic peripheral devices for a standard microprocessor having an appropriate clock means and control means and provided with a plurality of kinds of shift registers controlled by a clock means, two multiplexed serial/parallel multiplexers, a borrow detector, an auxiliary subtractor, an auxiliary adder, a delay register, and a switching element. The electronic device is formed by integrating all of the above components so that the device can simultaneously and synchronously execute modular multiplication, modular square, and modular power.



(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平7-253949

(43)公開日 平成7年(1995)10月3日

(51)Int.Cl.⁶

G 0 6 F 15/16
7/72

識別記号

3 3 0 D

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数28 F D (全 34 頁)

(21)出願番号 特願平5-326008

(22)出願日 平成5年(1993)11月30日

(31)優先権主張番号 1 0 3 9 2 1

(32)優先日 1992年11月30日

(33)優先権主張国 イスラエル (I L)

(31)優先権主張番号 1 0 4 7 5 3

(32)優先日 1993年2月16日

(33)優先権主張国 イスラエル (I L)

(31)優先権主張番号 1 0 6 9 2 3

(32)優先日 1993年9月6日

(33)優先権主張国 イスラエル (I L)

(71)出願人 593231656

フォートレス ユー アンド ティー リ
ミティド

イスラエル国, ピアーシェバ 84110, ピ
ー, オー, ボックス 844

(72)発明者 カーミ デビッド グレッセル

イスラエル国, モービル ポスト ネゲブ
85530, キブツ ウリム (番地なし)

(72)発明者 デビッド ヘンデル

イスラエル国, ラーナナ, ハシャロン ス
トリート 16

(74)代理人 弁理士 宇井 正一 (外4名)

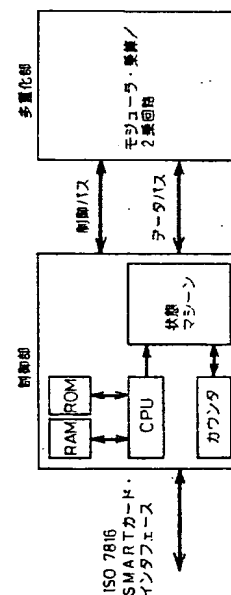
最終頁に続く

(54)【発明の名称】 モジュラ・乗算およびモジュラ・べき乗を遂行する超小形電子系装置ならびにその遂行方法

(57)【要約】

【目的】 本発明は、大きな数に対するモジュール・乗算およびべき乗を遂行するための超小形電子系装置ならびにその遂行方法に関し、モントゴメリの手法に基づきモジュラ・べき乗等の遂行に要求される時間を減らすことを目的とする。

【構成】 本発明は、適切なクロック手段および制御手段を有する標準のマイクロプロセッサに対するコンパクトな同期式の電子系超小形周辺機器からなり、各々が細分化される共に、切替制御可能であり、かつ、前記クロック手段により制御される複数種のシフトレジスタと、多重化され、かつ、直列／並列形の2つのみのマルチプレクサと、ボロー検出器と、補助的な減算器および加算器と、ディレイ・レジスタおよび切替素子とを備えており、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗を同時処理かつ同期方式により遂行するために、前述のすべての構成部品を集積化して形成する。



【特許請求の範囲】

【請求項1】 大きな数に対しモジュラ・乗算およびモジュラ・べき乗を遂行するための超小形電子系装置であって、

該超小形電子系装置は、適切なクロック手段および制御手段を有する標準のマイクロプロセッサに対するコンパクトな同期式の電子系超小形周辺機器からなり、

さらに、該超小形電子系装置は、

各々が細分化される共に、切替制御可能であり、かつ、前記クロック手段により制御される複数種のシフトレジスタ(B、SおよびN)と、

多重化され、かつ、直列／並列形の2つのみのマルチプレクサと、

ボロー検出器と、

補助的な減算器および加算器と、

ディレイ・レジスタおよび切替素子とを備えており、

前記超小形電子系装置は、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗を同時処理かつ同期方式により遂行するために、前記のすべての構成部品を集積化して形成することを特徴とする超小形電子系装置。

【請求項2】 前記超小形電子系装置が、ハードウェアの乗算、2乗およびべき乗に対し設計されたモントゴメリの方法をもとに展開されるような新奇かつ複合形で同期式のハードウェア装置により実現される請求項1記載の装置。

【請求項3】 前記超小形電子系装置が、モントゴメリの方法を展開することにより、並列動作方式に直列動作方式を取り入れた多数の同時処理と直列処理との複合形、すなわち、乗算、減算、加算、記憶形ディレイおよび 2^k による除算を遂行する装置として機能する請求項1記載の装置。

【請求項4】 前記超小形電子系装置が、モントゴメリの方法を展開することにより、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗のための多数の直列処理を遂行し、かつ、膨大な内部バスの使用を回避する請求項1記載の装置。

【請求項5】 前記超小形電子系装置が、モントゴメリの方法を展開することにより、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗のための多数の直列処理を遂行し、

前記超小形電子系装置は、一般の $1\mu\text{m}$ 技術を用いたSMARTカード用のISO7816の標準規格により規定されるマイクロチップ上に形成される程度に充分コンパクトである請求項1記載の装置。

【請求項6】 前記超小形電子系装置が、モントゴメリの方法を展開することにより、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗のための多数の直列処理を遂行し、

前記超小形電子系装置は、基本のアーキテクチャを変えることなく、特に、デュアルポート・アクセスのため

のメモリを再設計することなく、かつ、ファームウェアの要求が少ない状態で、1つの内部バスを備えた任意のマイクロプロセッサにより制御することが可能である請求項1記載の装置。

【請求項7】 前記超小形電子系装置が、マイクロプロセッサを使用してカスケード形の p 領域内での2乗および乗算の処理手順を規定し、

さらに、前記超小形電子系装置は、 n ビット長のシフトレジスタを含み、かつ、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗を遂行する多重化部を備え、該多重化部内に指数 E を記憶することが不要であるために、該多重化部による制御を簡単にし、また一方で、ほんのわずかな付加的なマイクロコントローラのROMコードのみを必要とする請求項1記載の装置。

【請求項8】 Bのレジスタが回転動作を行っている間に、オンザフライ方式により2乗の被乗数を用いて A_i のレジスタをロードする結果として、前記 A_i のレジスタを前記Bのレジスタにより再ロードする際に、マイクロコントローラによりBおよび／または $B-N$ の前の計算処理の最終値が取り出されるおそれが回避され、このために、該マイクロコントローラのRAMが節減され、かつ、2乗の繰り返し動作の各々において少なくとも n クロック分の実効的なクロック・サイクルを除去することが可能になる請求項1記載の装置。

【請求項9】 $Z/2^k$ が N よりも大きいか、または N に等しいかを決定し、たった1回の直列形の減算のみ行うような比較的小さい N オペランドを達成することにより、モントゴメリの方法による簡単な装置から、2つの記憶用レジスタおよび独立の直列形の減算処理が除去されると共に、 $Z/2^k - N$ に関する単一の直列形の検出を行うことが可能になる請求項1記載の装置。

【請求項10】 3つの同時乗算処理を遂行する際に2つの直列／並列形の乗算器のみが使用されるように、半並列形式で回路の同期をとり、このために、シリコンを用いた装置において全シリコン領域に対し直列／並列形の乗算器の占める面積の割合が40%に抑えられ、

3つの直列／並列形の乗算器の代わりに2つの乗算器のみを使用することにより、シリコン領域内の直列／並列形の乗算器のセルを二重化し、該二重化構造の乗算器のセルを動作させることにより、512ビットの乗算処理に必要な時間を従来の45%に節減することが可能になる請求項1記載の装置。

【請求項11】 k ビットのシフトレジスタからなる1つのデジタルのディレイ素子を使用して X の直列形の加算と乗算器(ML1)の直列結果との同期をとり、直列／並列形の乗算器の積または繰り返し処理が二重に記憶されることを防止する請求項1記載の装置。

【請求項12】 各々が k ビットのシフトレジスタからなる2つのデジタルのディレイ素子を使用して3つの

10

20

30

40

50

直列形の乗算を遂行し、該乗算は、 N を1つの因子としたときに $B \cdot A_i$ 、 $X \cdot J_0$ および $Y_0 \cdot N$ のように表される請求項1記載の装置。

【請求項13】 処理の流れの中で2つの独立した乗算動作、すなわち、 $X \cdot J_0$ および $Y_0 \cdot N$ を遂行することができるように、1つのデジタルのディレイ素子を使用して直列／並列形の乗算器(ML2)の乗算動作の同期をとる請求項1記載の装置。

【請求項14】 前記シフトレジスタ(B 、 S および N)が、 n ビット長または $n/2$ ビット長で構成され、 $n/2$ の長さのモジュールに対するべき乗が、 n ビット長のべき乗に対し必要であろうと思われる実効的なクロック・サイクル期間の $1/8$ より幾分少ない時間で遂行される請求項1記載の装置。

【請求項15】 オリジナルの検索因子 T で処理がなされる場合、全RSA記号のべき乗処理における ρ 領域内での乗算動作の回数が、半分近くに減少する請求項1記載の装置。

【請求項16】 必要に応じて前もって計算することを仮定した場合、オンザフライ方式により A のレジスタをロードし、かつ、オンザフライ方式により S のレジスタの内容の大きさを予測し、さらに、オンザフライ方式により一部のオペランドの同期をとることにより、 n ビットの数の乗算処理 $\rho(A \cdot B)N$ が、実効的な $m(n+2k)$ クロック・サイクルで完全に遂行される請求項1記載の装置。

【請求項17】 小規模のボー検出回路が付加され、かつ、制御メカニズムに簡単な付加物が付加されているようなモントゴメリの乗算処理に対し使用されるものと同一機器の同じレジスタを用い、第2のモードにおいて H パラメータの計算を行う請求項1記載の装置。

【請求項18】 ρ 領域内での乗算またはべき乗が公知のクロック・サイクルの処理で遂行されるように、すべての副処理過程および処理過程が、予め定められたクロック・サイクル数でそれぞれ実行され、このために、内部の条件設定用ブランチを使用することなく、カスケード形かつ自励式のカウンタ・メカニズムからなる簡便化された制御が可能になる請求項1記載の装置。

【請求項19】 モジュラ・乗算を遂行するための方法であって、被乗数 A 、乗数 B およびモジュロ N の各々が、 k ビット長の m キャラクタから構成され、乗数 B はモジュロ N よりも大きくない値であり、前記方法は、下記のステップを有しており、

第1のステップで、 H パラメータと、他のパラメータの少なくとも最下位のキャラクタ J_0 とを前もって計算し、かつ、該キャラクタ J_0 を k ビットのレジスタにロードし、

第2のステップで、前記乗数 B およびモジュロ N を、それぞれ対応する n ビット長のレジスタにロードし、こ

で、 $n=m \cdot k$ のように表され、

第3のステップで、 n ビット長のレジスタ S のビット値をすべて0にし、第4のステップで、 i 番目の繰り返し動作を m 回遂行し、ここで、 i は0から $m-1$ までの数であり、さらに、 i 番目の繰り返し動作の各々は、以下の動作を含み、

(a) 前記被乗数 A の i 番目のキャラクタ A_i を、 A_i のレジスタ手段から、レジスタおよびラッチ手段から選定された記憶手段へ転送し、

10 (b) $X=S(i-1)+A(i-1)*B$ により表される X の値を生成し、ここで、 $S(i-1)$ は S の更新された値であり、 S の更新は、次のように定義され、

①乗算手段に対し、 B のレジスタを周期的に右方向へシフトし、

②直列形式で B を A_i により乗算し、

③前記モジュール N を周期的に右方向へシフトし、

④ $S(i-1)$ が N よりも大きくない場合、 $(i-1)$ 番目の繰り返し動作の後に S のレジスタに記憶される値を $S(i-1)$ の更新された値として決定し、 $S(i-1)$ が N よりも大きい場合、直列形式で $S(i-1)$ から N を引くことにより得られる値を $S(i-1)$ の更新された値として決定し、さらに、この結果として得られる $S(i-1)$ の更新された値を設定し、

⑤ S のレジスタを周期的に右方向へシフトし、さらに、各ビット毎に、乗算 $A(i-1)*B$ を $S(i-1)$ の更新された値に加算し、

(c) $X(X_0)$ の最下位のキャラクタを J_0 により乗算し、 N および X が k クロック・サイクルだけ遅延されている間に、 $X_0 * J_0 \bmod 2^k$ の値を Y_0 のレジスタ手段に入れ、

30 (d) $Z=X+Y_0 * N$ の Z の値を計算し、この計算は、次のように行われ、

① N のレジスタに対し遅延かつ右方向へのシフトがなされた状態で Y_0 を N により乗算し、同時に、この乗算結果に対し、前述の周期的な右方向へのシフトがなされ、

② X を $Y_0 * N$ の値に加算し、

(e) Z の最下位のキャラクタを無視し、残りのキャラクタを S のレジスタに入れ、このときに、最後の繰り返し動作以外は、 $Z/2^k$ を入れることになり、

40 (f) 前述と同様の方法により $S(i-1)$ の更新された値を決定するために、各ビット毎に $Z/2^k$ と N とを比較し、

(g) 前記被乗数 A の i 番目のキャラクタ A_i を、前記の動作期間において、 A のレジスタ手段にロードし、第5のステップで、最後(m 回目)の繰り返し動作においては、 $Z/2^k$ の最下位のキャラクタを無視し、残りのキャラクタを、 $C \mp \rho(A * B)N$ として B のレジスタに入れ、

50 第6のステップで、前記第3および第4のステップを繰り返し、ここで、 C が N よりも大きい場合には、 C また

ビットを無視する工程、

9. 該ビットEのそれぞれに付いて、0か1かに関係なく、上記で定義された二乗方法による工程4及び5の操作を実行する工程であって、該被乗数と該乗数とが共に該レジスタBから派生されるものであり、且つ該モントゴメリー乗算器に於ける連続する特性値が該レジスタBからレジスタAに格納される工程、

10. 若し、べき指数Eに関する現在のビットが、1であるか、或いは1に過ぎない場合に、工程9の操作終了後、前記で定義された二乗方法に関して工程4及び5を実行し、その際、該被乗数がレジスタBの内容であり、且つ該乗数がベースA*である工程、及び

11. べき指数Eの全てのビットに付いて、工程8~10の操作が実行された後に、レジスタBの内容を前記オリジナルベースAで付加的に乗算し、 $D \leftarrow AE \bmod N$ としての最後の操作に付いての結果を該レジスタBに格納する工程とから構成されている事の特徴とするモジュラ・べき乗 $D = AE \bmod N$ を実行する方法。

【請求項28】 平均有効長が $n/2$ ビットである2つの数値に付いて従来の乗算を実行する方法で有って、該乗算方法は、該請求項19に於いて定義されている乗算方法により該数値に対してモジュラ・乗算処理を実行するもので有って、該モジュラ、N、は全てが“1”s”(ffffffffff...ffff)で構成されたn-ビット数で、J0から1に対応するものであり、被乗数をレジスタBに格納し且つ請求項1に於いて定義されている乗算方法に従ってAを取り扱うものであり、Nは、全て1によりプリローディングレジスタNの手段によるか、或いは一連の“ハード”1を出力する為のNを出力するマルチプレクサーをセットすることにより、該Nは全て1と成りうるものである事の特徴とする方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、複数種の素数および複合形素数モジュールからなるガロアの領域において、大きな数に対しモジュラ処理を遂行するための手法に関する。さらに詳しくいえば、本発明は、大きな数に対するモジュール・乗法およびモジュール・べき乗を遂行するための超小形電子系装置ならびにその遂行方法について言及するものである。このようなモジュール・乗算およびモジュール・べき乗は、公開キー暗号の認証や暗号化プロトコル(Protocol)に不可欠な演算を遂行する場合に適している。この種の演算は、小規模のマイクロプロセッサによっては通常の処理時間内に実行することが不可能である。

【0002】

【従来の技術、および、発明が解決しようとする課題】本発明は、インタリーピング方式によるモントゴメリ(Montgomery)の多倍精度の乗算方法」として知られている手続きを、ハードウェアによって遂行することに関

係する。このモントゴメリの乗算方法は、暗号化のソフトウェア指向システムに度々使用される。ここでは、モジュラ・べき乗を促進するために、唯一のオリジナルな方法が提供される。さらに、このモジュラ・べき乗を遂行するためのアーキテクチャを単純化し、かつ、通常使用されるような数の領域内でモジュラ・べき乗を遂行する装置の適用範囲を拡大するために、欠くことのできないプルーフが使用される。

【0003】上記の手続きに関する基本的な処理は、モントゴメリの方法論に基づきモジュラ・乗算を遂行するための手法に関連するような3つの公知の方法のいずれか一つにより実行される。これらの公知の方法の1つめは、P. L. モントゴメリ著の「試行除算を行わない方式のモジュラ・乗算(Modular Multiplication without trial division)」(計算の数学(Mathematics of Computation)、第44巻、519~521頁、1985年発行)に記載されている。なお、これ以降は、上記の1つめの方法は、単にモントゴメリの方法とよぶこととする。公知の方法の2つめは、S. R. デュッセ(Dusse)およびB. S. カリスキー ジュニア(Kaliski Jr.)著の「モトローラ社製のDSP 56000に向けられた暗号のライブラリ(A Cryptographic Library for the Motorola DSP 56000)」(90年代の欧州暗号化の議事録(Proc. Eurocrypt '90)、1990年にベルリンのシュプリング出版社(Springer-Verlag)により発行)に記載されている。なお、これ以降は、上記の2つめの方法は、単にデュッセの方法とよぶこととする。

【0004】上記の手続きをハードウェアにより遂行する場合、機密保護機構や、オンザフライ(On the Fly)式の加算、減算、および、けた移動が付加される。さらに、総合的な出力が不適切であるような処理が除去される。さらにまた、シリコンを利用した設計に基づき、比較的容易に遂行できるような手段が開発されて集積化される。この集積化により実現された装置は、実際には、8ビット、16ビットまたは32ビットの中央処理装置(CPU)の従装置として、内部のデータ/アドレスバスに付加される。

【0005】本発明に係る乗算/2乗機器は、簡単な同期形のけた送り式の設計がなされているために、これまでに達成された速度の何倍ものクロック速度で動作することが可能である。このクロック速度は、ボード上に実装されかつ不揮発性の記憶装置(メモリ)を支援するCPUにより実現される。乗算/2乗機器を用いた手法は、CPUのメモリのアーキテクチャにおける設計変更を必要としない。この種のアーキテクチャは、例えばフィリップ回路のように、並列形の乗算器とデュアルポートのメモリを用いることによって大きな数に対する高速のモジュラ・乗算を遂行する場合に規定されるものである。このモジュラ・乗算を遂行するための複数の

は $C-N$ が B にとって代わり、さらに、 H が A にとって代わることによって $P = \rho (C * H) N$ を計算し、第7のステップで、最後の繰り返し動作により得られる P の値を、 $A * B \bmod N$ と仮定することを特徴とする方法。

【請求項20】 n が、256と512との間の数から選定されるか、または、 k の倍数の増分から選定される請求項19記載の方法。

【請求項21】 前記被乗数 A と前記乗数 B とが同じ数である場合に、モジュラ・2乗およびモジュラ・乗算を遂行する請求項19記載の方法。

【請求項22】 $D = A^E \bmod N$ により表されるモジュラ・乗算およびモジュラ・べき乗を遂行するための方法であって、該方法は、複数の乗算処理および2乗処理を含む請求項19記載の方法。

【請求項23】 1. モジュラスをレジスタ N に格納する工程、

2. レジスタ S を0にセットする工程、

3. べき乗化されるべきベース A をレジスタ B に格納する工程、

4. べき指数 E をコンピューターのレジスタに格納する工程、

5. 該べき指数 E を左にシフトさせる工程、

6. 第1番目の1ビットに先行するそれらの0ビットを無視すると共に、該べき指数 E と第7及び8の操作を実行する為のビットに続く全てに対して、第1番目の1ビットを無視する工程、

7. 該ビットのそれぞれに付いて、0か1かに関係なく、上記で定義された乗算方法により該レジスタ B の内容を二乗すると同時に、該ベースの連続する特性値が該レジスタ B からレジスタ A に格納される工程、

8. 若し、べき指数 E に関する現在のビットが、1であるか、或いは1に過ぎない場合に、工程7の操作終了後該レジスタ B の内容を該ベース A で乗算する工程、及び
9. べき指数 E の全てのビットに付いて、工程6～8の操作が実行された後に、 $D = A^E \bmod N$ としての最後の操作に付いての結果を該レジスタ B に格納する工程とから構成されている請求項22に記載の方法。

【請求項24】 1. モジュラスをレジスタ N に格納する工程、

2. レジスタ S を0にセットする工程、

3. べき乗化されるべきベース A をレジスタ B に格納する工程、

4. べき指数 E をコンピューターのレジスタに格納すると共に、以下に定義する事前演算パラメータ T をコンピューターCPUに格納する工程、記載の方法。

5. 該べき指数 E を左にシフトさせる工程、

6. 第1番目の1ビットに先行するそれらの0ビットを無視すると共に、該べき指数 E と第7及び8の操作を実行する為のビットに続く全てに対して、第1番目の1ビ

ットを無視する工程、

7. 該ビットのそれぞれに付いて、0か1かに関係なく、請求項1で定義された乗算方法に関して工程4及び5を実行すると同時に、被乗数と乗数とがベース A であり、且つ該ベースの連続する特性値が該レジスタ B からレジスタ A に格納される工程、

8. 若し、べき指数 E に関する現在のビットが、1であるか、或いは1に過ぎない場合に、工程7の操作終了後、請求項1で定義された乗算方法に関して工程4及び5を実行し、その際、該被乗数がレジスタ B の内容であり、且つ該乗数がベース A である工程、及び

9. べき指数 E の全てのビットに付いて、工程7～8の操作が実行された後に、レジスタ B の内容を前記パラメータで付加的に乗算し、 $TD = A^E \bmod N$ としての最後の操作に付いての結果を該レジスタ B に格納する工程とから構成されている事を特徴とする請求項19による繰り返し操作を実行することにより、モジュラ・べき乗 $D = A^E \bmod N$ を実行する方法。

【請求項25】 コンピューターCPUと乗算回路を含む制御手段から構成されている請求項19記載の方法により、モジュラ・乗算を実行する装置であって、該乗算回路は、乗数としての n -ビットシフトレジスタ B 、モジュラスとしての n -ビットシフトレジスタ N 、本発明に於いて定義されている値 S としての n -ビットシフトレジスタ N 、被乗数としての k -ビットシフトレジスタ A_i 、本発明に於いて定義されている値 J_0 及び Y_0 としての k -ビットレジスタ手段、該レジスタ B の内容と該レジスタ A_i の内容とを掛け合わせる乗算手段、付加的な n -ビット乗算器手段及び加算、減算、多重化及び遅延手段とを含んでいる事を特徴とする装置。

【請求項26】 該 n -ビットレジスタとその他の構成部分との間の接続、及びラッチ回路以外の構成部分間の接続は1ビット接続である事を特徴とする請求項25記載の装置。

【請求項27】 1. べき指数 E をコンピューターの記憶手段に格納する工程、

2. モジュラスを前記レジスタ N に格納する工程、

3. 前記レジスタ S を0に設定する工程、

4. 前記した特許出願番号104753に記載された方法に従って、

$A * = \rho (AH) N$ の乗算操作を実行する工程、

(此処で、 A は、べき乗化されるべきオペランドであり H は、前記で定義した事前演算パラメータである。)

5. 該 $A *$ を該ベースレジスタ B に格納する工程、

6. 該ベースレジスタ B の内容に対して二乗演算操作を実行する工程、

7. 該べき指数 E を左にシフトさせる工程、

8. 第1番目の1ビットに先行するそれらの0ビットを無視すると共に、該べき指数 E と第9及び10の操作を実行する為のビットに続く全てに対して、第1番目の1

手法の3つめは、フィリップ社製の電子部品「83C852（条件付きのアクセス・アプリケーション用の機密保持機能付きの8ビット・マイクロコントローラ）」

（1990年8月にアインホーベン（Eindhoven）にて発表）により実行される方法である。なお、これ以降は、上記の3つめの方法は、単にフィリップの方法とよぶこととする。

【0006】上記のような基本的なアーキテクチャは、任意のマイクロコントローラの設計に際し集積化が可能であり、かつ、メモリへのマッピングが可能であるような機器に適用される。さらに、この種の機器はまた、コマンドおよびオペランドを常時ロードし、かつ、最終的な応答結果を取り出して転送することが必要なマイクロコントローラと並列に動作することも可能でなければならない。

【0007】このような要求に対する新奇な解は、2つの直列／並列式の乗算器のみを用いると共に、完全な直列形のパイプライン方式のアプローチを採用することである。このようなパイプライン方式のアプローチを採用することにより、シリコンの面積を節減することが可能になる。現在一般に用いられている周知の技術によれば、メモリ付きのマイクロコントローラを有すると共に上記の解を完全に満足するような装置を、 4×4 、 5×0 、2の大きさの電子系回路上に集積化することが可能である。このようにして得られる電子系回路は、ISO 7816の規格を満足するものである。ここで、ISOとは、国際標準化機構（International Organization for Standardization）の略号であり、認証カード、すなわち、集積回路カード（ICカード）において特定される。上記ISOの中のISO 7816は、下記の3つの部分から構成される。

【0008】①第1部…ISO 7816-1（物理特性）、1987年制定

②第2部…ISO 7816-2（接点の位置の寸法）、1988年制定

③第3部…ISO/IEC 7816-3（電子信号および通信プロトコル）、1989年制定

なお、これ以降は、これらの3つの部分を併せてISO 7816とよぶこととする。

【0009】本発明は、モントゴメリにより開示された数学的な革新事項に基づいて上記の新奇な解のアーキテクチャを実現することに向けられている。本発明では、後述するように、モジュラ・べき乗の遂行に要求される時間を、公知の処理方法およびモントゴメリの方法を用いた場合に必要な処理時間の半分の時間とほとんど変わらない値にまで減らすために、幾つかの変形や、改良や、機能的な方法が提供される。

【0010】ここで、本発明のモジュラ・乗算およびモジュラ・べき乗を遂行する装置および方法を述べる前に、一般的な計算の数学について大まかに説明すること

とする。

数学上の定義、一般的な法則および処理方法

素数および複合形基本モジュールからなる数の領域において、我々は、AおよびBを、それぞれ被乗数および乗数として定義する。さらに、通常は、AまたはBよりも大きい数としてNを定義する。ただし、場合によっては、Nは、Aよりも小さい数になり得る。さらに、A、BおよびNの各々を、 $m \times k$ （積の記号は、 \times の代わりに、 \cdot または $*$ で表すこともある） $= n$ ビット長のオペランドとして定義する。各kビットのグループは、キャラクタとよばれる。それゆえに、A、BおよびNの各々は、mキャラクタ長のオペランドから構成される。ここでは、モジュラ・乗算およびモジュラ・べき乗の最初の遂行過程の理解と、1ステップ毎にモジュラ・乗算およびモジュラ・べき乗を遂行する手続の説明を容易にするために、我々は、A、BおよびNの各々を、512ビット長（ $n = 512$ ）のオペランドとして定義する。さらに、kを32ビット長に設定する。この32ビットは、現在、乗算器のコストに見合ったkの長さともなせる。さらに、 $m = 16$ としてmの値を設定する。このmの値16は、一つのオペランドにおけるキャラクタの数であり、かつ、512ビットのオペランドに対する2乗ループまたは乗算ループの繰り返しの回数である。この場合、明らかに、いずれのオペランドも整数である。

【0011】さらに、我々は、モジュールの数の合同を表すために、記号“ \equiv ”を使用する。例えば、 $16 \equiv 2 \bmod 7$ と記載されている場合、16が2モジュロ7と合同であり、かつ、16を7で割ったときの剰余が2であることを意味する。また、 $Y \bmod N \equiv X \bmod N$ と記載されている場合、XおよびYの両方が、Nよりも大きい可能性がある。さらに、XおよびYが正であるときは、それぞれの剰余は同一の値になるであろう。さらに、Yが負の整数である場合に、Yの合同は $Y + uN$ で表されることに注意すべきである。この場合、Yの合同がNよりも小さい値であるならば、uは、Yの合同を正の値にするための最小の数として設定されるであろう。

【0012】さらに、我々は、より限定された意味における合同を表すために、記号“ \equiv ”を使用する。ここで述べる処理過程においては、各種の値は、度々、望ましい値か、または、望ましい値とモジュールとの和になる。例えば、 $X \equiv 2 \bmod 7$ と記載されている場合、Xは、2または9のいずれかに等しい。このときに、Xは、 $2 \bmod 7$ に対し限定された合同を有するものとして定義される。

【0013】さらに、 $X = A \bmod N$ と記載されて場合、我々は、AをNで割った場合の剰余としてXを定義する。例えば、 $3 = 45 \bmod 6$ のように表される。数の理論においては、モジュラ・逆数が基本的な概念になる。例えば、Xのモジュラ逆数は、 X^{-1} と表される。この場合、モジュラ逆数 X^{-1} は、 $XX^{-1} \bmod N = 1$ の関

係式により定義される。もし、 X の値が3に等しく($X=3$)、かつ、 N の値が13に等しければ($N=13$)、 X^{-1} の値は9になる($X^{-1}=9$)。すなわち、積 $3 \cdot 9$ を13により割った値は1になる。

【0014】この場合、参照すべきビット、キャラクタおよび全オペランドの値の最上位または最下位を表示するために、頭字語MSおよびLSがそれぞれ使用されることがある。本明細書中で、 N は、値 N 、および、この値 N を含むシフトレジスタの名前の両方を意味している。 A および N は、べき乗処理の全過程を通して一定の値である。さらに、 A は、べき乗処理がなされるべき数の値である。べき乗処理の最初の繰り返し動作においては、 B は A に等しい。 A はまた、累算された値が存在するレジスタの名前でもある。この場合、累算された値は、最終的に、べき乗処理の望ましい結果に等しくなる。 S は、一時的な値を示すと共に、値 S に対し限定された合同(\equiv)の関係を有する値が記憶されるようなレジスタを示す。 $S(i-1)$ は、 i 回目の繰り返し動作の始まりにおける S の値を意味する。 S_0 は、 $S(i)$ の値の最下位(LS)のキャラクタを示す。

【0015】ここで、我々は、 ρ 領域(この“ ρ ”はベクトルで表示すべきものであるが、電子出願の形式では、 ρ をベクトルにて表示することができないので、やむを得ず通常のギリシャ文字で表示することとする)における乗算 $\rho(A \cdot B)N$ の処理過程を簡単に説明する。なお、乗算 $\rho(A \cdot B)N$ の詳しい定義は、後ほど行うこととする。

【0016】この $\rho(A \cdot B)N$ 以外の記号は、算術計算の中で通常使用されるものである。

モントゴメリ方式のモジュラ・乗算

モジュラ・乗算 $A \cdot B \bmod N$ を遂行するための古典的なアプローチにおいては、積 $A \cdot B$ の剰余は、除算処理*

$$P \cdot 2^n = A \cdot B + Q \cdot N$$

【0020】この式(1)は、最下位の n ビットの値が0になるような 2^n ビット長の表現が可能であることを意味する。ここで、 $I \cdot 2^n$ が $I \bmod N$ と合同である($I \cdot 2^n \equiv I \bmod N$)と仮定する。この場合、 I はすべての奇数の N に対して存在する。前述の式(1)の両※

$$P \cdot I \cdot 2^n \equiv N; \text{ (ただし、} I \cdot 2^n \equiv I \bmod N \text{)} \quad (2)$$

【0022】また一方で、式(1)の左辺では、下記の式(3)に示すような合同の関係が導き出される。★

$$A \cdot B \cdot I + Q \cdot N \cdot I \equiv A \cdot B \cdot I \bmod N; \quad \text{(ただし、} Q \cdot N \cdot I \equiv 0 \bmod N \text{)} \quad (3)$$

【0024】この結果、前述の式(2)および式(3)より下記の式(4)が導き出される。☆

$$P \equiv A \cdot B \cdot I \bmod N \quad (4)$$

【0026】残念なことではあるが、この式(4)より、 ρ 領域の乗算が実行される度に寄生因子(寄生関数) I が導入されることがわかる。ここで、 ρ 演算子を下記の

*を利用することにより計算される。しかしながら、このような除算動作を実行することは、乗算動作を実行することよりも難しい。

【0017】モントゴメリのモジュラ・縮小法を用いることにより、上記の除算処理は、実質的に、前もって計算された定数を使用した乗算処理に置き換えられる。モントゴメリの関数 $\rho(A \cdot B)N$ は、 ρ 領域内で積 $A \cdot B$ の乗算モジュロ N を遂行する。 ρ 領域から通常のモジュールの領域への検索処理は、 $\rho(A \cdot B)N$ の結果と前もって計算された定数 H をもとに ρ を規定することにより遂行される。ここで、 P が $\rho(A \cdot B)N$ と合同である場合($P \equiv \rho(A \cdot B)N$)、 $\rho(A \cdot B)N$ は $A \cdot B \bmod N$ に等しくなる($\rho(A \cdot B)N = A \cdot B \bmod N$)。それゆえに、 ρ 領域での2つの乗算処理によって通常のモジュラ・乗算がなされることになる。

【0018】効果的なモジュラ・縮小法を使用する意図は、 n ビット長および $2n$ ビット長のオペランドに対する一連の乗算および除算動作を回避することにある。このような乗算動作および除算動作の回避は、元の値が n ビット長であってかつ最高値が n ビット長の最終結果を生成するようなオペランドに対し一連の乗算、加算および減算を実行することにより実現される。上記のようなモントゴメリの指針を証明するために、我々は、所定の A 、 B 、および奇数の N (この奇数のモジュールは、常に、単純な大きな素数かまたは複合形の大きな素数のいずれかである)に対し、次のような Q が最終的に存在することに注意すべきである。すなわち、 $A \cdot B + Q \cdot N$ が、最下位の n ビットの値が0になるような数になるという条件を満足する Q が存在することである。さらに詳しくは、このような条件を下記の式(1)に示す。

【0019】

【数1】

(1)

※辺に I を掛けることにより、式(1)の左辺では、下記の式(2)に示すような合同の関係が導き出される。

【0021】

【数2】

40★【0023】

★【数3】

☆【0025】

☆【数4】

式(5)のように定義する。

【0027】

【数5】

【0044】この式(13)により、定数Jが求まる。ここで、JはNのみの関数なので、前もって計算された定数となる。さらに、明らかなことではあるが、我々は、Nよりも小さい正の値のJを選定しなければならない。これまでの説明より当業者にとっては明らかなように、上記の処理過程において、所定のA、B、N、および、前もって計算された定数に対し、3つの乗算処理と、1つの加算処理と、最高の減算処理とを遂行することによって $\rho(A \cdot B)N$ が得られる。さらに、このようにして得られた結果と、同じような処理過程と、前もって計算された定数H（モジュールNの関数）とを用いることにより、 $A \cdot B \bmod N$ が求められる。この場合、AはBに等しくなるので、モジュラ・算術計算により2乗または乗算を行うための装置に対し、このような演算子を使用することが可能になる。

【0045】インタリービング方式によるモントゴメリのモジュラ・乗算 *

$$J_0 \equiv -N_0^{-1} \bmod 2^k \quad (J_0 \text{ は、} N \text{ が奇数のときに存在}) \quad (14)$$

【0048】ここで、前述のモントゴメリのインタリービング方式による縮小法を用い下記のステップ1)～ステップ5)を遂行することによって、次のような初期条件の下でm回の繰り返し動作の後に $\rho(A \cdot B)N$ が規定される。本発明の回路は、これらの複数のステップを並列方式により実行する。 ※

For $i = 1, 2 \dots m$:

$$1) \quad X = S(i-1) + A_{i-1} \cdot B$$

[A_{i-1} はAの($i-1$)番目のキャラクタ;

$S(i-1)$ は*i*回目の繰り返し動作の始まりにおけるSの値を示す]

value

$$2) \quad Y_0 = X_0 \cdot J_0 \bmod 2^k$$

(積 $X_0 \cdot J_0$ 中の最下位のkビット)

$$3) \quad Z = X + Y_0 \cdot N$$

$$4) \quad S(i) = Z / 2^k$$

(Z中の最下位のkビットは常に0であり、それゆえにZは常に 2^k により割り切れる)

$$5) \quad S(i) = S(i) \bmod N$$

(Nは、Nよりも大きい $S(i)$ から引かれる)

最終的に、最後の繰り返し動作において、

$$C = S(m) = \rho(A \cdot B)N \quad (15)$$

【0050】ここで、ステップ5)の除算処理は、Z中の最下位のkビットが常に0である場合の右方向へのkビットのけた移動（シフト動作）に相当する。または、除算処理回路に見られるように、Z中の最下位のkビットが単純に無視される。上記のように、最後の繰り返し動作の後に、式(15)が得られる。あるいは、必要に応じ

* 前述のセクションにおいては、すべてnビット長の複数のオペランド、および、 $2n+1$ ビットの記憶領域が要求される複数の計算結果に対して乗算を必要とするようなモジュラ・乗算の方法を述べてきた。ここで、さらに、モントゴメリのインタリービング方式による縮小法（既に記載済みのデュッセの論文に記述されている）を利用することにより、より短いオペランド、レジスタ、およびハードウェア乗算器を用いた乗算動作を実行することができる。この結果として、論理ゲートの数が比較的少ない電子装置による処理が可能になる。

【0046】さらに、kビットの乗算器を用いることにより、kビット長のキャラクタを定義することが容易に行える。この場合、nビット中にm個のキャラクタが存在する($m \cdot k = n$)。Jの最下位の文字として J_0 を定義することにより、下記の式(14)が導き出される。

【0047】

【数14】

※初期条件： $S(0) = 0$ （最初（1回目）の繰り返し動作の始まりにおけるSに対し限定された合同の関係を有する値）

【0049】

【数15】

て、Nを引いた後に式(15)が得られる。 $F = A \cdot B \bmod N$ を導き出すために、我々は、 ρ 領域における $\rho(C \cdot H)N$ の計算を実行しなければならない。

【0051】ここで、我々は、すべての $S(i)$ に対し、 $S(i)$ の値が $2N$ よりも小さいことを証明する（モントゴメリの証明には含まれていない）。ここでの

$$P \equiv A \cdot B \cdot I \pmod{N} \equiv \rho(A \cdot B) N \quad (5)$$

【0028】さらに、式(5)のPを、“ ρ 領域における * 演算することにより遂行される。
AとBとの乗算”とよぶこととする。 ρ 領域からの検索 【0029】
処理は、下記の式(6)に示すように、 $P \cdot H$ に対し ρ を* 【数6】

$$\rho(P \cdot H) N \equiv A \cdot B \pmod{N}; \quad (6)$$

【0030】この式(6)のような合同の関係におけるP ※【0031】
を式(4)のPで置き換えることにより、Hの値が導き出 【数7】
される。この経過を下記の式(7)に示す。 ※

$$\rho(P \cdot H) N \equiv (A \cdot B \cdot I)(H)(I) \pmod{N}; \quad (7)$$

($A \cdot B \cdot I \leftarrow P$; $H \leftarrow H$; I —すべての乗算動作は
寄生関数Iを発生する)

【0032】ここで、Hが、 I^2 の逆数と合同である場 ★【0033】
合、式(7)は有効となり、下記の式(8)が成立する。 ★ 【数8】

$$H \equiv I^{-2} \pmod{N} \equiv 2^{-2} \pmod{N} \quad (8)$$

(HはNの関数であり、Hパラメータとよばれる)

【0034】 $A \cdot B$ に対し ρ 演算子を規定するために、 ☆(9)が成立する。
前もって計算された定数Jを用いて下記のステップ1) 【0035】
からステップ5)までの処理を遂行することとする。こ 20 【数9】
の場合、最終的に、ステップ5)において、下記の式 ☆

- 1) $X = A \cdot B$
- 2) $Y = (X \cdot J) \pmod{2^n}$ (最下位のnビットのみ必要)
- 3) $Z = X + Y \cdot N$
- 4) $S = Z / 2^n$ (Jに対する条件を満たすために、Zが 2^n に
より割り切れるようにすることが必要となる)
- 5) $P \equiv S \pmod{N}$ (もし $S \geq N$ であれば、SからNが引かれる)

最終的に、ステップ5)において、

$$P \equiv \rho(A \cdot B) N \quad (9)$$

(Nを引いた後、必要な場合には、 $P = \rho(A \cdot B) N$)

【0036】これらの処理に続き、下記の式(10)が導き ◆【0037】
出される。 ◆ 【数10】

$$Y = A \cdot B \cdot J \pmod{2^n} \quad (n \text{ ビット中の最下位ビットのみ使用}) \quad (10)$$

【0038】さらに、下記の式(11)が導き出される。 * 【数11】

$$Z = A \cdot B + (A \cdot B \cdot J \pmod{2^n}) \cdot N \quad (11)$$

【0040】ここで、Zが 2^n (Zの最下位のnビット ※【0041】
が0でなければならない)により割り切れるためには、 【数12】
下記の式(12)に示す合同が存在することが必要である。 ※

$$(A \cdot B + (A \cdot B \cdot J \pmod{2^n}) \cdot N) \pmod{2^n} \equiv 0 \quad (12)$$

【0042】さらに、この式(12)のような合同が存在す ★ばならない。
るためには、 $N \cdot J \pmod{2^n}$ が-1と合同であること 【0043】
が必要である。すなわち、下記の式(13)が成立しなければ ★ 【数13】

$$J \equiv -N^{-1} \pmod{2^n} \quad (13)$$

処理過程に使用されるオペランドに対し、下記の式(16)のような3つの不等式が成立することに注意すべきである。

$$S(i-1) < N; \quad B < N \quad \text{and} \quad A_{i-1} < 2^k \quad (16)$$

【0053】（これらの不等式中の最初の2つは、 $S(i-1)$ および B が N に等しいかまたは大きい場合に、繰り返し動作の始まりにおいて、これらの $S(i-1)$ および B から N を引いたときに成立する。さらに、 2^k が、最上位 (MS) のビットが1であるような $k + ※$

* 【0052】
【数16】

※1ビット長の数であり、かつ、 A_{i-1} が k ビット長のオペランドである場合に、3つめの不等式が成立する。）
定義により、上記の式(17)が成立する。

【0054】
【数17】

$$S(i) = Z / 2^k \quad (\text{減算が可能な最後の部分における } S \text{ の値})$$

(17)

【0055】前述の一揃いの式において置換を行うことにより、下記の式(18)が成立する。

★ 【0056】
★ 【数18】

$$Z = S(i-1) + A_{i-1} \cdot B + (X_0 \cdot J_0 \bmod 2^k) N$$

(18)

【0057】ここで、式(18)における各要素の最高値を取り入れることにより、下記の式(19)のような Z に関する不等式が成立する。

☆ 【0058】
【数19】

☆20

$$\begin{aligned} Z &< (N-1) + (2^k - 1) \cdot (N-1) + (2^k - 1) \cdot N \\ &= 2^k N + 2^k N - N - 2^k \end{aligned} \quad (19)$$

【0059】この不等式より、下記の式(20)が確実に成立する。

◆ 【0060】
◆ 【数20】

$$Z < 2^k \cdot N + 2^k \cdot N \quad (20)$$

【0061】ここで、式(20)の不等式の両辺を 2^k で割ることにより、式(21)が得られる。

* 【0062】
* 【数21】

$$Z / 2^k < N + N \quad (21)$$

【0063】この式(21)の不等式より、いかなる場合にも、 $S(i)$ または B を調整するためには、 N を1回だけ減算すればよいことが証明された。

例1

インタリーピング方式によるモジュラ・乗算
16進法のモードの手動形計算機を使用することにより、インタリーピング方式によるモジュラ・乗算を利用した計算の有効性が容易に証明される。まず初めに、16進法のフォーマットを用いて次のように数を設定する。

【0064】 $N = a59$ (モジュロ)、 $A = 99b$ (乗数)、 $B = 5c3$ (被乗数)、 $n = 12$ (N のビット

長)、 $k = 4$ (乗数のビットの大きさであり、キャラクタの大きさでもある)、および、 $m = 3$ ($n = k \cdot m$)
さらに、 $J_0 = 7$ ($7 \cdot 9 \equiv -1 \bmod 16$)、および、 $H \equiv 2^{2 \times 12} \bmod a59 \equiv 44b$ が設定される。

【0065】ここで要求される結果は、 $F \equiv A \cdot B \bmod N \equiv 99b \cdot 5c3 \bmod a59 \equiv 375811 \bmod a59 = 220_{16}$ である。インタリーピング方式によるモジュラ・乗算を利用した計算の処理過程を下記に示す。

初期条件： $S(0) = 0$

【0066】
【数22】

初期条件 $S(0) = 0$

ステップ1 $X = S(0) + A_0 \cdot B = 0 + b \cdot 5c3 = 3f61$
 $Y_0 = X_0 \cdot J_0 \bmod 2^k = 7$
 $Z = X + Y_0 \cdot N = 3f61 + 7 \cdot a59 = 87d0$
 $S(1) = Z / 2^k = 87d \text{ (Nよりも小さい)}$

ステップ2 $X = S(1) + A_1 \cdot B = 87d + 9 \cdot 5c3 = 3c58$
 $Y_0 = X_0 \cdot J_0 \bmod 2^k = 8 \cdot 7 \bmod 2^k = 8$
 $Z = X + Y_0 \cdot N = 3c58 + 52c8 = 8f20$
 $S(2) = Z / 2^k = 8f2 \text{ (Nよりも小さい)}$

ステップ3 $X = S(2) + A_2 \cdot B = 8f2 + 9 \cdot 5c3 = 3ccd$
 $Y_0 = d \cdot 7 \bmod 2^k = b$
 $Z = X + Y_0 \cdot N = 3ccd + b \cdot a59 = aea0$
 $S(3) = Z / 2^k = aea, \text{ as } S(3) > N,$

$S(3) = aea - a59 = 91$

ゆえに、 $C = \rho(A \cdot B)N = 91,$

(22)

【0067】

* * 【数23】

(ρ 領域の検索は、 $\rho(C \cdot H)N$ を計算する
 ことにより遂行される：ここで、再び初期条件
 $S(0) = 0$ を設定する)

ステップ1 $X = S(0) + C_0 \cdot H = 0 + 1 \cdot 44b = 44b$
 $Y_0 = d$
 $Z = X + Y_0 \cdot N = 44b + 8685 = 8ad0$
 $S(1) = Z / 2^k = 8ad$

ステップ2 $X = S(1) + C_1 \cdot H = 8ad + 9 \cdot 44b = 2f50$
 $Y_0 = 0$
 $Z = X + Y_0 \cdot N = 2f50 + 0 = 2f50$
 $S(2) = Z / 2^k = 2f5$

ステップ3 $X = S(2) + C_2 \cdot H = 2f5 + 0 \cdot 44b = 2f5$
 $Y_0 = 3$
 $Z = X + Y_0 \cdot N = 2f5 + 3 \cdot a59 = 2200$
 $S(3) = Z / 2^k = 220,$

【0068】最終的に得られた値は、 $99b \cdot 5c3m$
 $oda59$ であり、前述の要求される結果に一致する。
 上記の乗算動作の有効性は、各ステップにおいて最下位
 の0の k ビットを無視する場合には、基本的に、上位の
 n ビットに対し 2^k を掛けることになることを我々が認
 識したときに、直観的に理解することができる。さら
 に、各ステップにおいて、乗数の i 番目の部分は、 2^{ik}
 により乗算される数である。このような乗算処理によ
 り、上記の部分は、 $S(i)$ と同じランクになる。

【0069】モントゴメリの機器における1つの乗算処
理過程内のモジュラ・縮小法

例えば、NISTのデジタル記号の標準化や、中国式
 の剰余定理を用いたモジュラ・べき乗のような多くの暗
 号化過程では、第2のモジュロよりも大きい（大抵の場
 合、2倍よりも大きい）数を減らすことが要求される。
 これらのモジュラ・縮小法は、インタリービング方式に
 よる1つのモジュラ・乗算処理過程において効果的に実
 行され得る。このモジュラ・乗算処理過程は、本発明に

初期条件: $S(0) = 0$, $A = t = 0af59b$.

$B = R = 141d$, $N = q = 2b13$

ステップ1 $X = S(0) + A_0 \cdot B = 0 + 9b \cdot 141d = c2d8f$

$Y_0 = X_0 \cdot J_0 \bmod 2^k = 8f \cdot e5 \bmod 2^k = eb$

$Z = X + Y_0 \cdot N = c2d8f + eb \cdot 2b13$

$= 23b800$

$S(1) = Z / 2^k \bmod N = 33b8$ (N よりも大きい)

$S(1) = 33b8 - 2b13 = 8a5$

ステップ2 $X = S(1) + A_1 \cdot B = 8a5 + f5 \cdot 141d$

$= 134866$

$Y_0 = X_0 \cdot J_0 \bmod 2^k = 66 \cdot e5 \bmod 2^k = 3e$

$Z = X + Y_0 \cdot N = 134866 + 3e \cdot 2b13$

$= 1db700$

$S(2) = Z / 2^k \bmod N = 1db7$

ステップ3 $X = S(2) + A_2 \cdot B = 1db7 + 0A \cdot 141d$

$= e6d9$

$Y_0 = d9 \cdot e5 \bmod 2^k = 1d$

$Z = X + Y_0 \cdot N = e6d9 + 1d \cdot 2b13 = 5c800$

$S(3) = Z / 2^k \bmod N = 5c8$

【0077】最終的に、前述のように、 $t \bmod q$ が5c8に等しいことが確認される。

べき乗

ここでは、D. ヌース (Nuth) 著による「コンピュータ・プログラミングの技術 (The Art of Computer Programming)」(半数学的アルゴリズム (Seminumerical Algorithms)、第2巻、アディソン・ウェスリ (Addison-Wesley) 社、読書協会 (Reading Mass)、1981年発行) による処理手順の方法をもとに、モジュラ・べき乗を遂行するための2乗および乗算の処理手順を説明す *

$$C = A^E \bmod N$$

【0080】この式(26)の計算処理過程を下記に示す。ここで、 $E(i)$ は、指数Eに対し2進法のビット表示を行ったときのi番目のビットを示す。このi番目のビットは、インデックスが1の最上位のビットで始まり、

40

*る。なお、これ以降は、上記の方法をヌースの方法とよぶこととする。

【0078】まず初めに、我々は、前のセクションにおいて定数を予め計算していると仮定する。我々はまた、本発明の装置が、 ρ 領域内で2乗および乗算の両方を遂行できると仮定する。このときに、我々は下記の式(26)のような計算を行うこととする。

【0079】

【数26】

(26)

インデックスがqの最下位のビットで終わる。

【0081】

【数27】

係る機器や、モントゴメリのアルゴリズムへの機能的な拡張を利用している。

【0070】前述の幾つかの例では、モジュロの長さであるようなオペランドの n ビットが、 N の正確な長さでもあることが暗に示されている。このような関係は、通常のべき乗および乗算に対しては、ひじょうに効果的である。しかしながら、数の大きさの縮小が必要な場合には、第2の定数 $I^{-1} = 2^n \bmod N$ を使用することが有効である。この第2の定数は、所定の数により乗算されるモントゴメリの数値を縮小することが要求される場合、1つの乗算処理の規模を最小のレベルにまで縮小するものである。上記の定数 I^{-1} は、定数 H を計算する場合（ H パラメータの計算に関するセクションを参照のこと）と同様のメカニズムによって計算される。さらに詳しくいえば、この I^{-1} の計算は、除数レジスタの最上位*

*のビットにおいて1のビット値が存在するように、除数のオペランドの最上位の部分にモジュール N を配置することにより実行される。ここで、けた移動/試行減算の回数は、明らかに $n \div I - L$ でなければならない。ただし、 L は、 N が関係するビットの数である。この場合、 I^{-1} は、 L ビット長のオペランドになることに注意すべきである。

【0071】上記の前提条件を証明するために、まず初めに、我々は、 $A \cdot B \bmod N$ ($\rho(A \cdot B)N$) に対するモントゴメリの乗算処理によって $A \cdot B \cdot I \bmod N$ と合同の関係を有するものが生成されることを繰り返し述べる。 B が I^{-1} に等しい（ $B = I^{-1}$ ）と仮定した場合、下記の式(24)が成立する。

【0072】

【数24】

$$\rho(A \cdot I^{-1})N = A \cdot I^{-1} \cdot I \bmod N = A \bmod N \quad (24)$$

【0073】例2

インタリーピング方式によるモントゴメリの縮小法

t がモジュロ q に縮小され得ること（ $t \bmod q$ ）を証明するために、乗数レジスタの長さが、 q の長さよりも大きくなるように設定する。ここで、最初に記憶される t は、24ビットの長さを有する。

【0074】さらに、ワードの長さ（機器の乗数の大きさ）が8ビットであるとし、かつ、次のような変数を仮定する。

$n = 24$ 、 $k = 8$ 、 $t = 0af59b$ 、 $q = 2b13$ 、

および、 $R = I^{-1} = 2^{24} \bmod q = 141d$

単純な除算の計算処理を用いて比較を行うことにより、 $t \bmod q$ が $5c8$ に等しいこと（ $t \bmod q = 5c8$ ）がわかる。このような処理過程を下記に示す。

【0075】この場合、一回のモントゴメリの乗算処理において縮小および検索が遂行されることに注意すべきである。

【0076】

【数25】

```

a)  B = A
    FOR j = 2 TO q
a)  B ≡ ρ (B · B) N
b)  B ≡ ρ (B · H) N (上記のステップ a), b) は、
      B ≡ B2 mod N に相当する)
    IF E(j) = 1 THEN
a)  B ≡ ρ (B · A) N
b)  B ≡ ρ (B · H) N (上記のステップ a), b) は、
      B ≡ B · A mod N に相当する)

```

【0082】各ステップから次のステップに移行する際に、BがNに等しいかまたは大きい場合には、常に、BからNが引かれる。最後の繰り返し動作の後に、Bの値は、 $A^E \bmod N$ に対し限定された合同の関係を有するようになる ($B \equiv A^E \bmod N$)。本発明の回路を用いてモジュラ・べき乗を遂行する場合に、より効果的に利用され得るような特許権を有する複数のプロトコルが存在する。ここで、我々は、本発明で述べる方法において、通常のべき乗処理の速度が2倍になるような2つの暗号化プロトコルを挙げることにする。

【0083】1つめの暗号化プロトコルは、R. L. リベスト (Rivest) 他著の「デジタル記号および公開キー暗号化システム (A Method for Obtaining Digital Signatures and Public Key Cryptosystems)」(ACM 委員会 (Comm. of the ACM)、第21巻、120~126頁、1978年発行)に記載されている。なお、これ以降は、上記の1つめのプロトコルを、RSAの方法とよぶこととする。2つめの暗号化プロトコルは、W. ディフィー (Diffie) およびM. E. ヘルマン (Hellman) 著の「暗号手法に関する新しい指針 (New Directions in Cryptography)」(情報理論のIEEE議事録 (IEEE Trans. on Inform. Theory)、VOL. IT-22、644~654頁、1976年発行)に記載されている。なお、これ以降は、上記の2つめのプロトコルを、ディフィーヘルマンの方法とよぶこととする。これらの2つの方法においては、一定の指数を用いて大部*

*分の難しいべき乗が遂行される。

【0084】次のセクションで述べる方法 (ρ 領域からの検索のために効果的な方法) は、一定の指数を使用することによって計算処理に必要な計算時間を減らすことに言及する。この方法が用いられた場合、前述のステップ b)' のべき乗処理過程 (すべての $\rho (A \cdot B) N$ の乗算処理) が取り除かれる。さらに、べき乗処理に関するq回目の繰り返し動作の後に得られるBの最終値が、モントゴメリの ρ 領域において、前もって計算された定数Tにより乗算される。

【0085】上記の暗号化プロトコルに従事する人々にとっては、中国式の剰余定理 (前述のヌースの文献中に記述されている) を用いた回路によってRSAの記号を運用することにより、計算時間が70%より小さい値にまで節減され得ることは、明らかなことである。

ρ 領域からの検索のために効果的な方法

ここでは、前述のセクションのべき乗処理および乗算処理のプロトコルが改善され得る。さらに、繰り返し動作がなされる間に、新たに前もって計算された定数Tを取り入れることにより、 ρ 領域内での乗算の回数を減らすことが可能になる。この場合、Tは、モジュロNおよび指数Eの関数である。このような方法を、下記の処理過程に示す。

【0086】

【数28】

$$T = (2^{\Sigma})^E \bmod N = (1^{-1})^E \bmod N$$

ここで、 $\Sigma = 2q^{-1} + E \bmod 2q^{-1}$

また、

qは、Eに関連するビットの数を示す (最後の0の部分は無視)

【0087】この場合、モジュラ・べき乗は、下記の手順にて遂行される。

【0088】

【数29】

初期条件: $B = A$

FOR $j = 2$ TO q

$B \leftarrow \rho(B \cdot B)N$

IF $E(j) = 1$ THEN

$B \leftarrow \rho(B \cdot A)N$

END FOR

$B \leftarrow \rho(B \cdot T)N$

$n = 4$ および $E = 5 = 0101$ と仮定し、さらに、 q (E の最後の
0 を無視した後) を 3 に設定すると、

$E(1) = 1; E(2) = 0; \text{and } E(3) = 1$ 、

次のように、 T が前もって計算される

$$T = (2^n)^E \bmod N = (1^{-1})^E \bmod N$$

$$\begin{aligned} \Sigma &= 2q^{-1} + E \bmod 2q^{-1} = 2^{3-1} + 5 \bmod 2^{3-1} \\ &= 4 + 1 = 5 \end{aligned}$$

ゆえに、

$$T = 1^{-5} \bmod N$$

【0092】さらに、下記の処理過程を遂行する。

※【数31】

【0093】

※

初期条件:

$B = A$

$j = 2, E(2) = 0$

$B \leftarrow \rho(B \cdot B)N = A^2 \cdot 1 \bmod N$

$j = 3, E(3) = 1$

$B \leftarrow \rho(B \cdot B)N = B^2 = A^4 \cdot 1^2 \cdot 1 \bmod N$

$B \leftarrow \rho(B \cdot A)N = A^4 \cdot 1^3 \cdot A \cdot 1 \bmod N$

最終的に、

$$\begin{aligned} B \leftarrow \rho(B \cdot T)N &= A^5 \cdot 1^4 \cdot 1^{-5} \cdot 1 \bmod N \\ &= A^5 \bmod N \end{aligned}$$

【0094】 A^E を計算するために、次のようなステップが続けて実行される場合には、パラメータ T の導入が回避され得る。ここで、我々は、前もって計算されたモントゴメリの定数が存在すると仮定し、かつ、本発明の装置が、 P 領域内で2乗および乗算の両方の処理を遂行すると仮定することにより、下記の計算を実行する。

【0095】 $C = A^E \bmod N$

この場合、 $E(j)$ は、指数 E に対し2進法のビット表示を行ったときの j 番目のビットを示す。この j 番目のビットは、インデックスが1の最上位のビットで始まり、インデックスが q の最下位のビットで終わる。奇数の指数に対しては、次のような処理過程によりべき乗を

*【0089】ここで、各ステップから次のステップに移行する際に、 B が N に等しいかまたは大きい場合には、常に、 B から N が引かれることを再び仮定する。さらに、 ρ 領域内でのすべての乗算は、同じ因子 I によるモジュラ・乗算 (例えば、 $\rho(X \cdot Y) = X \cdot Y \cdot I \bmod N$) に相当する点に再び注意すべきである。

【0090】例3

この例3は、 $A^E \bmod N$ の計算における T の有用性を証明し、かつ、 T の定義を明らかにするためのものである。この例3の処理過程を下記に示す。

10 【0091】

【数30】

*【数30】

遂行することができる。

40 【0096】

【数32】


```

A ← ρ (A · H) N
B ← A
FOR j = 2 TO q - 1
    B ← ρ (B · B) N
    IF E(j) = 1 THEN
        B ← ρ (B · A) N
ENDFOR
B ← ρ (B · A) N
C ← B

```

* 【0097】各ステップから次のステップに移行する際に、BがNに等しいかまたは大きい場合には、常に、BからNが引かれる。最後の繰り返し動作の後に、Bの値は、 $A^E \bmod N$ に対し限定された合同の関係 ($B \equiv A^E \bmod N$) を有するようになる。そして、最終値としてCが得られる。

【0098】また一方で、偶数の指数に対しては、前述の最後のステップが下記の式(33)によって置き換えられる。

10 【0099】
【数33】

*

$$B \leftarrow \rho (B \cdot I) N \quad (B \leftarrow \rho (B \cdot A) N \text{ に取って代わる}) \quad (33)$$

【0100】さらに、ここでの処理過程をより明確にするために、下記のような具体例を掲載する。

※ 【0101】

※ 【数34】

```

E = 1011 → E(1) = 1; E(2) = 0; E(3) = 1; E(4) = 1;
A1011 mod N を求める; q = 4
A ← ρ (A · H) N = A I-1 I = A I-1 mod N
B ← A
for j = 2 to q
    B ← ρ (B · B) N より A2 (I-1)2 · I = A2 · I-1
                                が導き出される
E(2) = 0;   B ← A2 · I-1
j = 3       B ← ρ (B · B) N = A2 (I-1)2 · I
                                = A4 · I-1
E(3) = 1   B ← ρ (B · A) N
                                = (A4 · I-1) (A I-1) · I = A5 · I-1
j = 4       B ← ρ (B · B) N = A10 · I-2 · I = A10 · I-1
E(4) が奇数なので、最後の乗算はAに基づいて遂行される。
この結果、寄生関数 I-1 が除去される
B ← ρ (B · A) = A10 · I-1 · A · I = A11
C ← B

```

【0102】Hパラメータの計算

Hパラメータは、モントゴメリの領域内での計算に際し不可欠なものであり、かつ、一定の値である。ある種のプロトコルを用いた場合、Hは、比較的大きなコンピュータで予め計算されるような定数になる。あるいは、他のプロトコルを用いた場合、Hは、より有効な定数を計算する際に使用される第1段階のパラメータであるような有用性のある定数にもなり得る。この点に関しては、前述のセクションを参照されたい。

★

$$H = 2^{2n} \bmod N$$

(35)

【0105】この式(35)は、Hパラメータが通常の除算動作の剰余であることを意味する。この場合、最上位の

★ 【0103】通常の通信においては、Hが前もって計算されることを仮定している。しかしながら、幾つかのプロトコル、例えば、RSAにおけるランダムな通信の際の記号の暗号化に対しては、本発明の装置、例えば、SMARTカードを使用してHを計算することも必要である。Hパラメータは、下記の式(35)により計算される。

【0104】
【数35】

1ビットと、これに続く最下位のビット値0の2nビット (2n+1ビット長のオペランド) とから構成される

ビット列が、モジュールの基数 N により除算されることになる。ビット値1の1ビット、および、ビット値0のビット列からなる被除数に対し除数 N による2進法の除算を遂行することは、 N による順次の試行減算を実行することに相当する。すなわち、上記の除算処理は、最上位の $n+1$ ビットが N よりも大きい場合に、残りの試行過程の被除数から N を引くことに相当するものである（下記の例を参照のこと）。

【0106】この場合、元の被除数は $2n+1$ ビット長であるが、除算処理により生成される残りの試行過程の*10

*被除数は、明らかに、 $n+1$ ビット長を超えることはない。さらに、最下位のけたが0になることも明らかである。例えば、次のような例を挙げる。すなわち、 $N=11_{10}=1011_n$ （したがって、 N のビット長は、4になる、したがって、 $n=4$ ）のときに、 H を求める例を挙げる。

【0107】除算の基数を2として、長い除算を手動により実行する。

【0108】

【数36】

	<u>1 011</u>	
1011	11 0000 0000	
	<u>1011</u>	減算成功
0101	0	← 1回目の丸めの結果
	<u>101 1</u>	減算失敗
101	00	← 2回目の丸めの結果
	<u>10 11</u>	減算成功
10	010	← 3回目の丸めの結果
	<u>1 011</u>	減算成功
0	1110	← 4回目の丸めの結果
	<u>1011</u>	減算成功

5回目（ $(n+1)$ 回目）の丸めの結果⇒0011= H （10進数の3＝剰余）

【0109】最終的に、 $H=3_{10}$ であることが確認された。上記の除算処理では、 $n+1$ 回の試行減算が存在する。さらに、試行減算により得られる被除数もまた、 $n+1$ ビット長になることに注意すべきである。このような減算処理の手順は、後述の本発明のハードウェアの説明箇所ですくしく述べることにする。

【0110】

【課題を解決するための手段、および、作用】本発明は、大きな数に対しモジュラ・乗算およびモジュラ・べき乗を遂行するための超小形電子系装置に係り、適切なクロック手段および制御手段を有する標準のマイクロプロセッサに対するコンパクトな同期式の電子系超小形周辺機器からなる。

【0111】さらに、本発明この超小形電子系装置は、各々が細分化される共に、切替制御可能であり、かつ、前記クロック手段により制御される複数種のシフトレジスタと、多重化され、かつ、直列／並列形の2つのみのマルチプレクサと、ボロー検出器と、補助的な減算器および加算器と、ディレイ・レジスタおよび切替素子とを備えている。

【0112】このような超小形電子系装置は、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗を同時処理かつ同期方式により遂行するために、前述のすべての構成部品を集積化して形成する。好ましくは、本発明

の超小形電子系装置は、ハードウェアの乗算、2乗およびべき乗に対し設計されたモントゴメリの方法をもとに展開されるような新奇かつ複合形で同期式のハードウェア装置により実現される。

【0113】さらに、好ましくは、本発明の超小形電子系装置は、モントゴメリの方法を展開することにより、並列動作方式に直列動作方式を取り入れた多数の同時処理と直列処理との複合形、すなわち、乗算、減算、加算、記憶形ディレイおよび 2^k による除算を遂行する装置として機能する。さらに、好ましくは、本発明の超小形電子系装置は、モントゴメリの方法を展開することにより、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗のための多数の直列処理を遂行し、かつ、膨大な内部バスの使用を回避し得る。

【0114】さらに、好ましくは、本発明の超小形電子系装置は、モントゴメリの方法を展開することにより、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗のための多数の直列処理を遂行し、一般の $1\mu m$ 技術を用いたSMARTカード用のISO7816の標準規格により規定されるマイクロチップ上に形成される程度に充分コンパクトである。

【0115】さらに、好ましくは、本発明の超小形電子系装置は、モントゴメリの方法を展開することにより、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき

30

40

50

乗のための多数の直列処理を遂行し、基本のアーキテクチャを変えることなく、特に、デュアルポート・アクセスのためのメモリを再設計することなく、かつ、ファームウェアの要求が少ない状態で、1つの内部バスを備えた任意のマイクロプロセッサにより制御することが可能である。

【0116】さらに、好ましくは、本発明の超小形電子系装置は、マイクロプロセッサを使用してカスケード形の p 領域内での2乗および乗算の処理手順を規定する。さらに、前記超小形電子系装置は、 n ビット長のシフトレジスタを含み、かつ、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗を遂行する多重化部を備えている。この多重化部内に指数 E を記憶することが不要であるために、この多重化部による制御を簡単にし、また一方で、ほんのわずかな付加的なマイクロコントローラのROMコードしか必要としない。

【0117】さらに、好ましくは、本発明の超小形電子系装置は、 B のレジスタが回転動作を行っている間に、オンザフライ方式により2乗の被乗数を用いて A_i のレジスタをロードする結果として、前記 A_i のレジスタを前記 B のレジスタにより再ロードする際に、マイクロコントローラにより B および/または $B-N$ の前の計算処理の最終値が取り出されるおそれが回避される。

【0118】このために、このマイクロコントローラのRAMが節減され、かつ、2乗の繰り返し動作の各々において少なくとも n クロック分の実効的なクロック・サイクルを除去することが可能になる。本発明の構成によれば、 $Z/2^k$ が N よりも大きいか、または N に等しいかを決定し、たった1回の直列形の減算のみ行うような比較的小さい N オペランドを達成することにより、モン

トゴメリの方法による簡単な装置から、2つの記憶用レジスタおよび独立の直列形の減算処理が除去されると共に、 $Z/2^k - N$ に関する単一の直列形の検出を行うことが可能になる。

【0119】さらに、本発明の構成によれば、3つの同時乗算処理を遂行する際に2つの直列/並列形の乗算器のみが使用されるように、半並列形式で回路の同期をとっている。このために、シリコンを用いた装置において全シリコン領域に対し直列/並列形の乗算器の占める面積の割合が40%に抑えられる。

【0120】さらに、本発明の構成によれば、 k ビットのシフトレジスタからなる1つのデジタルのディレイ素子を使用して X の直列形の加算と乗算器の直列結果との同期をとって、直列/並列形の乗算器の積または繰り返し処理が二重に記憶されることが防止される。さらに、本発明の好ましい実施態様においては、シフトレジスタが、 n ビット長または $n/2$ ビット長で構成され、 $n/2$ の長さのモジュールに対するべき乗が、 n ビット長のべき乗に対し必要であろうと思われる実効的なクロック・サイクル期間の $1/8$ より幾分少ない時間で遂行

される。

【0121】さらに、本発明の好ましい実施態様においては、オンザフライ方式により A のレジスタをロードし、かつ、オンザフライ方式により S のレジスタの内容の大きさを予測し、さらに、オンザフライ方式により一部のオペランドの同期をとることにより、 n ビットの数の乗算処理 $p(A \cdot B) \cdot N$ が、実効的な $m(n+2k)$ クロック・サイクルで完全に遂行される。

【0122】さらに、本発明の好ましい実施態様においては、小規模のボロー検出回路が付加され、かつ、制御メカニズムに簡単な付加物が付加されているようなモン

トゴメリの乗算処理に対し使用されるものと同じ機器の同じレジスタを用い、第2のモードにおいて H パラメータの計算を行うことができる。本発明のモジュラ・乗算を遂行するための方法においては、被乗数 A 、乗数 B およびモジュロ N の各々が、 k ビット長の m キャラクタから構成され、乗数 B はモジュロ N よりも大きくない値に設定される。

【0123】前記の方法は、下記のステップにより実行される。第1のステップで、 H パラメータと、他のパラメータの少なくとも最下位のキャラクタ J_0 とを前もって計算し、かつ、このキャラクタ J_0 を k ビットのレジスタにロードし、第2のステップで、前記乗数 B およびモジュロ N を、それぞれ対応する n ビット長のレジスタにロードし、ここで、 $n=m \cdot k$ のように表され、第3のステップで、 n ビット長のレジスタ S のビット値をすべて0にし、第4のステップで、 i 番目の繰り返し動作を m 回遂行し、ここで、 i は0から $m-1$ までの数であり、さらに、 i 番目の繰り返し動作の各々は、以下の動作を含み、(a)前記被乗数 A の i 番目のキャラクタ A_i を、 A_i のレジスタ手段から、レジスタおよびラッチ手段から選定された記憶手段へ転送し、(b) $X=S(i-1)+A(i-1) \cdot B$ により表される X の値を生成し、ここで、 $S(i-1)$ は S の更新された値であり、 S の更新は、次のように定義され、

- ①乗算手段に対し、 B のレジスタを周期的に右方向へシフトし、
- ②直列形式で B を A_i により乗算し、
- ③前記モジュロ N を周期的に右方向へシフトし、
- ④ $S(i-1)$ が N よりも大きくない場合、 $(i-1)$ 番目の繰り返し動作の後に S のレジスタに記憶される値を $S(i-1)$ の更新された値として決定し、 $S(i-1)$ が N よりも大きい場合、直列形式で $S(i-1)$ から N を引くことにより得られる値を $S(i-1)$ の更新された値として決定し、さらに、この結果として得られる $S(i-1)$ の更新された値を設定し、
- ⑤ S のレジスタを周期的に右方向へシフトし、さらに、各ビット毎に、乗算 $A(i-1) \cdot B$ を $S(i-1)$ の更新された値に加算し、
- (c) $X(X_0)$ の最下位のキャラクタを J_0 により乗

算し、NおよびXがkクロック・サイクルだけ遅延されている間に、 $X_0 * J_0 \bmod 2^k$ の値をY₀のレジスタ手段に入れ、(d) $Z = X + Y_0 * N$ のZの値を計算し、この計算は、次のように行われ、

①Nのレジスタに対し遅延かつ右方向へのシフトがなされた状態でY₀をNにより乗算し、同時に、この乗算結果に対し、前述の周期的な右方向へのシフトがなされ、

②XをY₀ * Nの値に加算し、

(e) Zの最下位のキャラクタを無視し、残りのキャラクタをSのレジスタに入れ、このときに、最後の繰り返し動作以外は、 $Z / 2^k$ を入れることになり、(f) 前述と同様の方法によりS(i-1)の更新された値を決定するために、各ビット毎に $Z / 2^k$ とNとを比較し、

(g) 前記被乗数Aのi番目のキャラクタA_iを、前記の動作期間において、Aのレジスタ手段にロードし、第5のステップで、最後(m回目)の繰り返し動作においては、 $Z / 2^k$ の最下位のキャラクタを無視し、残りのキャラクタを、 $C \neq \rho(A * B)N$ としてBのレジスタに入れ、第6のステップで、前記第3および第4のステップを繰り返し、ここで、CがNよりも大きい場合には、CまたはC-NがBにとって代わり、さらに、HがAにとって代わることによって $P = \rho(C * H)N$ を計算し、第7のステップで、最後の繰り返し動作により得られるPの値を、 $A * B \bmod N$ と仮定する。

【0124】さらに、本発明の方法によれば、被乗数Aと乗数Bとが同じ数である場合に、モジュラ・2乗およびモジュラ・乗算が遂行される。さらに、本発明の方法によれば、 $D = A^E \bmod N$ により表されるモジュラ・乗算およびモジュラ・べき乗が遂行される。さらに、本発明の方法は、

1. モジュラスをレジスタNに格納する工程、
2. レジスタSを0にセットする工程、
3. べき乗化されるべきベースAをレジスタBに格納する工程、
4. べき指数Eをコンピューターのレジスタに格納する工程、
5. 該べき指数Eを左にシフトさせる工程、
6. 第1番目の1ビットに先行するそれらの0ビットを無視すると共に、該べき指数Eと第7及び8の操作を実行する為のビットに続く全てに対して、第1番目の1ビットを無視する工程、
7. 該ビットのそれぞれに付いて、0か1かに関係なく、上記で定義された乗算方法により該レジスタBの内容を二乗すると同時に、該ベースの連続する特性値が該レジスタBからレジスタAに格納される工程、
8. 若し、べき指数Eに関する現在のビットが、1であるか、或いは1に過ぎない場合に、工程7の操作終了後該レジスタBの内容を該ベースAで乗算する工程、及び
9. べき指数Eの全てのビットに付いて、工程6～8の操作が実行された後に、 $D \neq A^E \bmod N$ としての最

後の操作に付いての結果を該レジスタBに格納する工程とから構成されている。

【0125】さらに、本発明の方法は、

1. モジュラスをレジスタNに格納する工程、
 2. レジスタSを0にセットする工程、
 3. べき乗化されるべきベースAをレジスタBに格納する工程、
 4. べき指数Eをコンピューターのレジスタに格納すると共に、以下に定義する事前演算パラメータTをコンピューターCPUに格納する工程、
- 記載の方法。

【0126】5. 該べき指数Eを左にシフトさせる工程、

6. 第1番目の1ビットに先行するそれらの0ビットを無視すると共に、該べき指数Eと第7及び8の操作を実行する為のビットに続く全てに対して、第1番目の1ビットを無視する工程、
7. 該ビットのそれぞれに付いて、0か1かに関係なく、請求項1で定義された乗算方法に関して工程4及び5を実行すると同時に、被乗数と乗数とがベースAであり、且つ該ベースの連続する特性値が該レジスタBからレジスタAに格納される工程、
8. 若し、べき指数Eに関する現在のビットが、1であるか、或いは1に過ぎない場合に、工程7の操作終了後、請求項1で定義された乗算方法に関して工程4及び5を実行し、その際、該被乗数がレジスタBの内容であり、且つ該乗数がベースAである工程、及び
9. べき指数Eの全てのビットに付いて、工程7～8の操作が実行された後に、レジスタBの内容を前記パラメータで付加的に乗算し、 $TD = A^E \bmod N$ としての最後の操作に付いての結果を該レジスタBに格納する工程とから構成されている事を特徴とする請求項19による繰り返し操作を実行することにより、モジュラ・べき乗 $D \neq A^E \bmod N$ を実行する。

【0127】さらに、本発明の方法は、コンピューターCPUと乗算回路を含む制御手段から構成されている請求項19記載の方法により、モジュラ・乗算を実行する装置であって、該乗算回路は、乗数としてのn-ビットシフトレジスタB、モジュラスとしてのn-ビットシフトレジスタN、本発明に於いて定義されている値Sとしてのn-ビットシフトレジスタN、被乗数としてのk-ビットシフトレジスタA_i、本発明に於いて定義されている値J₀及びY₀としてのk-ビットレジスタ手段、該レジスタBの内容と該レジスタA_iの内容とを掛け合わせる乗算手段、付加的なn-ビット乗算器手段及び加算、減算、多重化及び遅延手段とを含んでいる。

【0128】さらに、本発明の方法は、該n-ビットレジスタとその他の構成部分との間の接続、及びラッチ回路以外の構成部分間の接続は1ビット接続である。さらに、本発明の方法は、

1. ベキ指数Eをコンピュータの記憶手段に格納する工程、
2. モジュラスを前記レジスタNに格納する工程、
3. 前記レジスタSを0に設定する工程、
4. 前記した特許出願番号104753に記載された方法に従って、 $A * = p (AH) N$ の乗算操作を実行する工程、（此处で、Aは、べき乗化されるべきオペランドでありHは、前記で定義した事前演算パラメータである。）
5. 該 $A *$ を該ベースレジスタBに格納する工程、
6. 該ベースレジスタBの内容に対して二乗演算操作を実行する工程、
7. 該べき指数Eを左にシフトさせる工程、
8. 第1番目の1ビットに先行するそれらの0ビットを無視すると共に、該べき指数Eと第9及び10の操作を実行する為のビットに続く全てに対して、第1番目の1ビットを無視する工程、
9. 該ビットEのそれぞれに付いて、0か1かに関係なく、上記で定義された二乗方法による工程4及び5の操作を実行する工程であって、該被乗数と該乗数とが共に該レジスタBから派生されるものであり、且つ該モンゴメリー乗算器に於ける連続する特性値が該レジスタBからレジスタA_iに格納される工程、
10. 若し、べき指数Eに関する現在のビットが、1であるか、或いは1に過ぎない場合に、工程9の操作終了後、前記で定義された二乗方法に関して工程4及び5を実行し、その際、該被乗数がレジスタBの内容であり、且つ該乗数がベース $A *$ である工程、及び
11. べき指数Eの全てのビットに付いて、工程8～10の操作が実行された後に、レジスタBの内容を前記オリジナルベースAで付加的に乗算し、 $DY = A^E \bmod N$ としての最後の操作に付いての結果を該レジスタBに格納する工程とから構成されている。

【0129】さらに、本発明の方法は、平均有効長が $n/2$ ビットである2つの数値に付いて従来の乗算を実行する方法で有って、該乗算方法は、該請求項19に於いて定義されている乗算方法により該数値に対してモジュラ・乗算処理を実行するもので有って、該モジュラ、N、は全てが“1”s (ffffff.....fff) で構成された n -ビット数で、J0から1に対応するものであり、被乗数をレジスタBに格納し且つ請求項1に於いて定義されている乗算方法に従ってAを取り扱うものであり、Nは、全て1によりプリローディングレジスタNの手段によるか、或いは一連の“ハード”1を出力する為のNを出力するマルチプレクサーをセットすることにより、該Nは全て1と成りうるものである。

【0130】

【実施例】本発明のモジュラ・乗算およびモジュラ・べき乗を遂行する超小形電子系装置ならびにその遂行方法は、添付図面（図1～図9）を参照しながら、本発明の

好ましい実施例を具体的に説明することにより、より良く理解されるであろう。これらの添付図面は、本発明の装置を全体的に理解するために必要な複数種の論理的概念を示すものである。すべての場合において、クロック信号に従い回路が動作する。そして、リセット信号がある場合には、このリセット信号は、回路を零の状態にすることを目的としている。

【0131】以下、図1～図9の添付図面を参照しながら、本発明の実施例を詳細に説明する。図1は、本発明の一実施例の装置構成を示すブロック図である。ここでは、本発明の装置が集積化されたモノリシック回路のブロック図が例示されている。図1において、多重化部（MULT部と記載されることもある）は、本発明の基礎となるハードウェア装置を備える。状態マシンは、多重化部の回路を駆動するための制御部を構成する。ROM（読み出し専用回路）部は、すべて不揮発性メモリ（ROMおよびEPROM）から構成される。このROM部には、SMARTカードを制御するためのプログラム、高信頼性の3つのグループからなる公開キー、および、多重化部や状態マシンを駆動するためのプログラムが格納されている。RAM（ランダムアクセス回路）部は、一時的に存在するオペランドを記憶するための揮発性メモリから構成される。この種のオペランドとして、べき乗処理がなされる予定のメッセージ、暗号化されるべき公開キー、多重化部に転送されるべきデータ等が挙げられる。CPU（中央処理装置）は、実際は、8ビットまたはそれより大きなビットの内部バスを有するような任意のマイクロコントローラである。

【0132】図2は、本発明の一実施例のモジュラ・乗算回路を示すブロック図である。この場合、モジュラ・乗算回路は、モジュラ・2乗およびモジュラ・べき乗を実行するために用いられる。図2において、参照番号10、11および12は、B、SおよびNのレジスタをそれぞれ構成するような n ビット長（ $n = k \cdot m$ ）の3つのレジスタを示すものである。これらのレジスタの各々には、乗数の値Sとモジュロの値がロードされる。上記レジスタは、好ましくは、2つの $n/2$ レジスタに分割されている。さらに、上記レジスタは、好ましくは、NおよびBのレジスタに対し、 k ビットの最下位ビット部分を含む。マルチプレクサ13、14および15は、それぞれ、上記レジスタの前部に配置される。この場合、もし、これらのマルチプレクサが細分化されて個々の部品として形成されているならば、各マルチプレクサが、各々のレジスタ前部に配置される。さらに、図2のブロック図に示すように、3つのレジスタは、直列形式でロードされるように意図されている。しかしながら、並列形式によるロードも可能である。

【0133】16、17および18は、いずれも k ビット長であり、かつ、 A_i 、 J_0 、および Y_0 の値をそれぞれ受け入れる3つのレジスタを示すものである。レジ

スタ16、17は、それぞれ、直列ロード／並列出力形のシフトレジスタと、直列および並列ロード／並列出力形シフトレジスタである。レジスタ18は、好ましくは、直列入力／並列出力形シフトレジスタである。これらのレジスタの内容は、それぞれ、構成要素21、22を介して乗算手段19、20により処理されるように意図されている。これらの構成要素21、22は、好ましくは、kビットのラッチである。これらの構成要素21、22がラッチである場合、これらの構成要素21、22は、kビットのバスを通してレジスタ16、17および18からロードされる。また一方で、上記の構成要素21、22がレジスタである場合、これらの構成要素21、22は、1ビットの接続部を通して直列形式でロードされ得る。

【0134】参照番号24、25、25'、26、36、37および38は、マルチプレクサを示している。乗算手段（乗算器）19、20は、直列入力A、並列入力B、および直列出力を有する乗算手段か、または、その他の直列／並列入力、および直列出力を有する乗算手段である。マルチプレクサ38は、通常の数の領域内で乗算処理を実行するために、モジュロNのビット値を強制的にすべて1（オール1）にするものである。

【0135】参照番号27、28、29、30および31は、1ビットの全加算／減算手段または半加算／減算手段を示している。この内、31は、全加算／減算手段を示している。参照番号32、33および34は、デジタル信号を遅延させることが可能なkビットかつkクロック・サイクルのディレイ手段を示している。これらのディレイ手段は、アナログ素子またはデジタル素子のいずれによっても構成され得るが、アナログ素子により構成するのが好ましい。35は、ボロー検出器を示している。このボロー検出器は、2ビットのラッチ／記憶手段である。図2からわかるように、本発明の装置は、例えば512ビットのような大きい数を取り扱うように意図されているにもかかわらず、わずかな数のkビットのバスをオプションで持っている以外には、バスを備えていない。このために、本発明の装置では、ハードウェアの節減が図れる。B、SおよびNのレジスタが、n/2ビットの部分有する場合、本発明の装置は、256ビットの数に対し乗算動作およびべき乗動作を遂行するために使用することが可能である。このために、本発明では、装置を使用する際の柔軟性を備えるという利点が生ずる。

【0136】図3は、本発明の一実施例の特殊なモジュラ・乗算回路を示すブロック図である。ここでは、本発明の一実施例のモジュラ・乗算回路を論理セルにより構成している。図3において、オペランドは、直列の接続部DIを介し、 A_i のラッチと、 J_0 のレジスタと、Bのレジスタと、Nのレジスタとに供給される。そして、オペランドによる処理結果は、直列の接続部DOを介

し、BのレジスタまたはNのレジスタから取り出される。

【0137】信号Xは、Bと A_i とSとの積 $B \cdot A_i \cdot S$ のビットの流れを合計した結果（和）に相当する（SとBの積は、Nよりも小さいと仮定する）。信号 Y_0 は、 J_0 とXの積 $J_0 \cdot X$ における最下位のkビットの流れに相当する。信号Zは、 Y_0 とNの積 $Y_0 \cdot N$ と、Xとの和に相当する。ここでは、Z中の最下位のkビットはすべて0なので、この最下位のkビットは無視される。この結果、最上位のnビットのみが、SまたはBに対し直列に供給される。

【0138】ボロー検出器は、 $Z/2^k$ の値がNより大きいかなんかを検出するための論理回路である。減算器（通常、Subと略記される、図3では、単に減算と記す）1および減算器2は、BおよびSの値がNよりも大きい場合は、常に、BおよびSのビットの流れからNのビットの流れを減算するように動作する。

【0139】加算器（通常、Addと略記される、図3では、単に加算と記す）1および加算器2は、Xの流れとZの流れを生成するために、ビットの流れを加算するように動作する。ディレイ素子（図3では、単にディレイと記す）1およびディレイ素子2は、シフトレジスタから構成される。これらのディレイ素子は、数学的処理の同期をとるための記憶手段を提供するために必要なものである。

【0140】図3では、クロックの制御は図示していない。ここでは、クロックは、状態マシンから供給されると仮定している。このクロックの供給は、前述の直列入力／直列出力形の論理回路の1つからデータを送り出したりこれらの論理回路の1つにデータを提供したりしなければならぬときは、いつでも行われる。他の制御、すなわち、マルチプレクサのアドレス制御や、ラッチ転送信号の制御等もまた、詳しく開示していない。なぜならば、これらの制御は、本明細書に含まれる説明事項より、当業者にとっては明らかであるからである。

【0141】さらに、図2および図3の装置が、どのようにして本発明の乗算の方法に関する複数の動作を遂行するかは、当業者にとっては明らかであろう。しかしながら、これらの複数の動作のタイミング関係は、念のために、次の図4に示す。図4は、本発明の一実施例による繰り返し動作（イテレーション）と乗算動作との間の時間的な関係を示す図である。この図においては、本発明の一実施例による実効的かつ連続的なクロック・サイクルにおいて遂行されるようなすべての各種動作が、図式的に示されている。この場合、 $n=512$ 、および、 $m=16$ に設定される。このような設定条件は、暗号化技術においては、比較的一般的な条件である。前述の図3に例示された実施例に従って本発明を実施する場合、 $n=256$ の条件下で本発明を実行するために、図4と同じ装置が使用される。

【0142】図4においては、一連の各種動作が、実効的なクロック・サイクルの関数として図示されている。この実効的なクロック・サイクル（実効クロック）に関しては、横軸に目盛りがふられている。各種動作の始まり、および、すべての繰り返し動作前のタイミングでは、B、SおよびNの値が、それぞれ対応するレジスタにロードされる。上記の繰り返し動作は、本発明の乗算の方法の一部をなす。Aの最初のキャラクタもまた、対応するレジスタにロードされる。kクロック・サイクルの期間において繰り返し動作が始まるや否や、BおよびSのレジスタの内容のシフト動作が行われる。n+kの実効的なクロック・サイクルの期間においてXの値が発生する。最初のkクロック・サイクルは、 X_0 の値を取り入れることにより占有される。最初の実効的なkクロック・サイクルの期間において、 Y_0 の値が取り入れられる。次の実効的なn+kクロック・サイクルの期間において、乗算器20に既に取り入れられているXの値がシフトするか、または、このXの値がディレイ素子34により遅延された後に加算器31に取り入れられる。Nの値は、3つの異なる時間位相にて使用される。初めの位相は、SおよびBを更新するために使用される。第2番目の位相は、実効的なkクロック・サイクルの遅延期間の後に、 Y_0 による乗算を遂行するために使用される。第2番目の位相は、2回目の実効的なkクロック・サイクルの遅延期間の後に、SまたはBの次の値がどのようにして更新されるかを検知するために使用される。同様の実効的なn+kクロック・サイクルの期間において、Zが計算され、かつ、 $Z/2^k$ が計算される。最初の実効的なkクロック・サイクルの始まりのタイミングで、 A_i のロードが始まる。さらに、繰り返し動作が連続している間は、 A_i のロードも続けて行われる。 $Z/2^k$ の最終値は、最初の実効的な2kクロック・サイクルの期間後のnクロック・サイクルの期間において、S（またはB）のレジスタに取り入れられる。

【0143】図5は、直列／並列乗算器のセルの構成を示す回路図である（この回路図の作成に際しては、専門の技術に精通している技術陣の助けを借りているが、彼らは、本発明に関係する直列／並列乗算器のセル構成の研究に関しては関与していない）。これらの複数のセルの各々は、後述の図6に示すような乗算器（通常、MP Lと略記される）を備える。

【0144】図6は、8ビットの直列／並列乗算器の構成を示す回路図である。この直列／並列乗算器は、符号のない直列／並列乗算器の乗算動作に対しブース（Booth）の乗算アルゴリズムを実行する。図3の乗算器（通常、MLと略記される、図3では、単に乗算と記す）1および乗算器2に示すような直列／並列乗算器は、kビット長である。この場合、MSセル、すなわち、最上位ビットのセルが退化していることに注意すべきである。並列の8ビットの被乗数が、XIの接続部に入力され

る。さらに、nビット長の直列の乗数がYの接続部に入力される（乗数の最上位の1ビットの後に、最初に現れる最下位のkビットの列はすべて0である）。さらに、乗算器による乗算結果である積は、出力側の接続部MOにおいて、最下位のkビットが最初に現れ、最上位のビットが最後に現れる。この場合、積の全体は、n+kビット長になる。

【0145】図7は、直列加算器の構成を示す回路図である。ここでは、Aの接続部とBの接続部に現れる2つのビットの流れを加算するための直列加算器が例示されている。この直列加算器においては、出力側の接続部Sにおいて、ビットの流れの和が出力される。図7においては、最下位のビットが最初に入力される。さらに、mビット長のオペランドに対する出力の流れは、m+1ビット長になる。m回の実効的なクロック・サイクルの最後の部分では、CIの出力は、数のビット列中のm+1番目のビットに相当する。

【0146】図8は、直列減算器の構成を示す回路図である。ここでは、Aの接続部とBの接続部に現れる2つのビットの流れの差を出力するための直列減算器が例示されている。この直列減算器においては、出力側の接続部Dにおいて、ビットの流れの差が出力される。図7においては、最下位のビットが最初に入力される。さらに、mビット長のオペランドに対する出力の流れは、mビット長になる。m回の実効的なクロック・サイクルの最後の部分では、BIの出力は、数のビット列中のm+1番目のビットに相当する。同様に、このBIの出力は、ボローを表示するためのボロー表示手段として機能する。

【0147】図9は、Hパラメータを計算するためのアーキテクチャを示すブロック図である。ここでは、nビット長のモジュールNに対しHパラメータを計算するためのハードウェア構成が例示されている。このような動作モードの間では、nビット長のモジュールに対し、Nのレジスタがn+1回だけ回転動作を遂行する。この回転動作は、Sのレジスタの回転動作に同期した状態で行われる。この場合、Sのレジスタは、減算器1を介し、最下位のビットの遅延と一緒に回転動作を遂行する（最下位のビット値0は、最初のクロック・サイクルにおいてマルチプレクサ（M2 1; 1）に挿入される）。ボロー検出器は、回転動作が完了するような最後のタイミングで、次の丸めにおいてSの流れからNが引かれるか否かを認識する。さらに、ボロー検出器は、次の丸めに応じて前回の減算マルチプレクサを切り替える。

【0148】前述のように、図1は、本発明の方法を遂行するための装置をブロック図の形で表したものである。図1の装置における制御部は、下記の構成要素を備える。

(1) 完備した形のCPU（中央処理装置）

(2) カウンタ

(3) 状態マシン

さらに、CPUは、不揮発性メモリおよび揮発性メモリを有する。これらの不揮発性メモリおよび揮発性メモリの幾つかは、乗算処理過程に使用され得る。さらにまた、上記CPUは、回路内のモジュールの計算機能ブロックを制御する。

【0149】さらに詳しくいえば、上記CPUは、下記の機能を有する。

(1) ホストと交信すること

(2) チップにデータをロードし、かつ、チップからデータを取り出すこと

(3) 回路に対し一連の数学的動作を遂行するように指示すること

(4) 他の暗号化システムおよび非暗号化システムにตอบสนองしてデータ処理動作を遂行すること

カウンタは、実際の状態マシンに対するアドレスを生成する。

【0150】状態マシンは、アドレスを復号化し、多重化部(MULT部)に対する複数種の制御信号を生成する。これらの制御信号は、多重化部に対し、 $\rho(A \cdot B)N$ 変換の計算を実行するために必要とされる適切な動作手順を遂行するように指示する(ここで、AはBに等しい)。図3は、本発明の物理的な形態(多重化部)を実施するためのハードウェアの装置をブロック図の形で表したものである。さらに、図3は、本発明に係る特許により保護されるべきアーキテクチャの幾つかの概念に焦点を絞る際の補助手段となるように意図されている。図3のブロックは、同時に、モントゴメリのモジュラ・乗算にて既述したような式(1)~(5)により規定された手順を遂行する。さらに、図3のブロックは、同期クロックを変えることなく、かつ、限定された合同の関係を有するSおよびBの変換を行うことなく上記の手順を遂行する。このセクションにおいて、我々は、定数

(Nの関数) J_0 およびHが前もって計算されることを仮定している。図3の回路は、 $\rho(A \cdot B)N$ を遂行する。この回路の機能を利用することにより、この回路は、次の計算を行うために使用され得る。

【0151】(1) $B \cdot A \bmod N$

(2) $B^2 \bmod N$

ただし、いかなる場合でも、BはNよりも小さくなくてはならない。ここで、 $C = B \cdot A \bmod N$ を遂行する手順を詳細に説明する。

(1) まず、プロセッサが、オペランドBをBのレジスタに前もってロードする。同様に、プロセッサは、オペランドNをNのレジスタに前もってロードする。

【0152】(2) 多重化部内の回路が次のSの値を計算し始める度に、回路は、次の A_j を前もってロードすることを(フラグを立てることにより)CPUに知らせる。S(m)回の繰り返し動作の後に、Bに対し限定

された合同の関係を有する数がBのレジスタに残る。

(3) 多重化部が、 $F = \rho(B \cdot H)N$ を計算する。ここで、プロセッサが H_j のキャラクタの処理手順を前もってロードする場合を除けば、Hは、上記のステップ(1)および(2)に記載された手順において、前もって計算された定数である(プロセッサが A_j のキャラクタを前もってロードする場合も同じことがいえる)。

【0153】また一方で、 $C = B^2 \bmod N$ を遂行する手順を詳細に説明する。

10 (1) まず、Bのレジスタが、Bに対し限定された合同の関係を有することがわかっている数を保持していると仮定する。さらに、Nのレジスタが、モジュールNを保持していると仮定する(2乗処理においては、一般にいえることである)。ここで、多重化部は、 B_0 と B_0 の最下位のキャラクタでもって A_j のレジスタを予めロードしておくことにより、2乗処理を進めることができる。

【0154】(2) $B = \rho(B \cdot B)N$ の計算処理は、上記の乗算動作における第2のステップ(ステップ(2))と同様の処理過程で進行する。ただし、Bのレジスタが回転動作を行っている場合に、 B_j のキャラクタの連続するローディング動作が、Bのレジスタから直列式かつオンザフライ方式になされるときは、この限りではない。

【0155】(3) 必要な場合には、上記の乗算動作における第3のステップ(ステップ(3))と同様の $\rho(B \cdot H)$ の計算を行う。当業者にとっては明らかなことなので、発明者は、直列/並列形乗算器および一般の構成部品が本発明そのものの一部を構成することは、敢えて主張しない。今後の説明は、一般に普及している標準の論理セルを使用していることを明確にするためになされるものである。ただし、論理セルの幾つかは、それほど度々一般に使用されていないかもしれない。ここで図示したゲート構成は、本発明の証明のために例示しているにすぎない。熟練された技術者は、これらの論理セルを最適化するであろう。

【0156】オペランドA、BおよびNは、いずれもnビット長であり、kビット長のキャラクタからなるm個のグループにより構成される。それゆえに、 $n = k \cdot m$ が成り立つ。k=32のハードウェア装置においては、mは、8ビットまたは16ビットの2進数のビット長になる。

乗算器1および乗算器2

これらの乗算器(ML)は、符号のない乗算動作に対しブースの乗算アルゴリズムを実行する。この場合、並列のオペランドは、kセル(ビット)長になっており、直列のロードされるオペランドは、任意の要望されるビット長になっている。

【0157】各々の直列/並列乗算器は、k-1のMPLセルからなる(図5参照)。MSビットに相当する最上位のセルは、ANDゲートのみから構成される。各々

のMPLセルは、Yの直列の入力ビットと、XIの並列の入力ビットとの乗算動作を行う。さらに、この乗算動作により、前段のMPLユニットの直列出力と、それ自身の前回のサイクルのキャリー出力ビットが生成される。上記のMPLセルは、これらの出力結果を合計する。

【0158】図5からわかるように、各々のMPLセルは、2ビットの乗算加算器である。MPLセルのブロックは、XIの入力ビットと、Yの直列の入力ビットとの*

$$DO = (DI + CI + XI \cdot Y) \bmod 2$$

【0160】このようにして記憶されたキャリー出力COは、次のサイクルに対するキャリー入力CIになる。このキャリー出力COは、ブーリアン (Boolean) の和により、下記の(38)により表される。

加算器1および加算器2

ここで使用される各々の加算器 (Add) は、Dフリップフロップからなる単純な1ビットの全加算器である。この加算器は、次のクロック・サイクルで出力されるキャリービットを記憶するために使用される (図7参照)。

【0161】図7に示すように、2つの入力A、Bは、前回のクロック・サイクルからのキャリー入力CIと一緒に加算される。この結果、モジュロ2の和が生成される。この和は、出力信号Sを取り出すために、Dフリップフロップに記憶される。加算器がリセットされたときは、キャリービットは0になる。

減算器1、減算器2および減算器3

図8に示すように、減算器 (Sub) の各ブロックは、前段のボローを記憶するためのDフリップフロップからなる全減算器である。このブロックは、前述の加算器のブロックとほぼ同じ構成になっている。ただし、減算器においては、Aの流れからBの流れを直列に引く点が加算器と異なる。

【0162】ディレイ素子1、ディレイ素子2およびディレイ素子3

これらのディレイ素子 (Delay) は、k-1ビットの連結された状態の記憶素子から構成される。これらのディレイ素子は、数学的处理において各種のオペランドの同期をとるために使用される。これらの同期動作は、回路を説明すれば明らかになるであろう。

【0163】 A_i 、 J_0 、および Y_0

これらのブロックは、kビット長の直列入力/並列出力形シフトレジスタである。この場合、kビットの入力ビットは、直列に入力される。kビットの実効的なクロック・サイクル期間の後に、これらのkビットは、並列形式で出力側に現れる。

【0164】図2においては、細い線が直列の1ビットの導体線を示し、太い線が並列のkビットの導体線を示している。

$M4 \quad 1; x$ 、 $M3 \quad 1; x$ 、および $M2 \quad 1; x$

これらのブロックは、1ビット出力のマルチプレクサで

*乗算動作を行う。さらに、このブロックは、DI (データ入力) からの乗算結果と、前回のサイクルからのCI (キャリー入力) との加算処理を行う。最終の結果として、DO (データ出力) と、次のサイクルに対するCO (キャリー出力) が得られる。このキャリー出力COは、Dフリップフロップに記憶される。データ出力DOは、下記の式(37)により表される。

【0159】

【数37】

(37)

ある。 $M4 \quad 1; x$ は、4つの入力から1つの出力を取り出すものである。 $M3 \quad 1; x$ は、3つの入力から1つの出力を取り出すものである。 $M2 \quad 1; x$ は、2つの入力から1つの出力を取り出すものである。 x は、特定の構成部品に対する明瞭なインデックスを示している。

$B(0:k-1)$ 、 $B(k:n1-1)$ 、 $B(n1:n2)$ 、 $S(0:n1-1)$ 、 $S(n1:n2)$ 、 $N(0:k-1)$ 、 $B(k:n1-1)$ 、および $B(n1:n2)$

これらのブロックは、シフトレジスタである。比較的ビット長の長いレジスタのビット列の大きさおよび位置が、括弧 () 内の数字によって示されている。例えば、 $X(s:t)$ は、 $t-s+1$ ビット長のシフトレジスタである。ここで、 s は、レジスタ $X(s:t)$ の最初のビットのインデックスであり、 t は、レジスタ $X(s:t)$ の最後のビットのインデックスである。例えば、 $B(0:511)$ は、次のような3つの比較的短いかスケード接続形のレジスタから構成される。すなわち、 $B(0:31)$ 、 $B(32:255)$ 、および $B(256:511)$ から構成される。

【0165】 $n1$ は、一般に、 $n/2$ (例えば、256) に等しい。 $n1$ は、 k の倍数でなければならない。 $n2$ は、 $n-1$ に等しい。 k は機器のキャラクタの長さ、すなわち、直列/並列乗算器の大きさである。したがって、最初の処理過程においては、次の値が予測される。

【0166】 $n1=256$ 、 $n2=511$ 、 $n=512$ 、および $k=32$

ラッチ1およびラッチ2

これらの2つのラッチは、kビットのレジスタである。これらのラッチは、乗算器内の並列データを保持するために使用される。このようなラッチの動作により、乗算処理において単一のクロックを用いた並列変換が可能になる。

【0167】多重化部 (MULT部) の動作...p領域内での乗算およびべき乗

説明を簡単にするために、我々は、レジスタ内のデータが実際に移動するクロック・サイクルのみを指定することとする。すなわち、我々は、このようにデータが移動

20

30

40

50

するサイクルを実効的なクロック・サイクルと定義する。

$\rho(A \cdot B)N$ の乗算

第1段階：初期ローディング

この段階では、DIを介して下記のレジスタがロードされる。

- 【0168】(1) J_0 のレジスタに J_0 をロード (CPUにより前もって計算される)
- (2) BのレジスタにBをロード
- (3) NのレジスタにNをロード
- (4) A_2 のレジスタにA、 A_0 の最初のキャラクタをロード

同時に、ステップ(2)において、レジスタSに対しビット値0がロードされる。

【0169】これらの5つのレジスタに所定の値をロードした後に、符号のな2つの直列／並列乗算器ML1およびML2と、直列加算器Ad1およびAd2と、直列減算器Sub1、Sub2およびSub3とがリセットされる。

第2段階：B・ A_0 の繰り返し動作の実行

レジスタ A_i にロードされたデータ A_0 は、ラッチ1に転送される。レジスタBは、周期的に右方向へのシフト動作を行う。繰り返し動作の始まりにおいては、ボロー検出器2の制御信号はビット値0になっている。それゆえに、Bの内容は、Sub1を通過しても変化しないままである。さらに、Bの内容は、ML1において A_0 により乗算される。レジスタBの出力は、変化しない状態で、その入力される。

【0170】このような乗算の結果は、Ad1において、レジスタSの内容に直列に加算される。最初の繰り返し動作のときは、レジスタSの内容はすべて0である。この動作により、前述のようにXが生成される。上記の処理過程が進行している間に、CPUは、Aの次のキャラクタである A_1 をラッチ1に予めロードしておく。

【0171】さらに、 J_0 のレジスタからラッチ1に J_0 がロードされる。XがML2に直列に入力され、 J_0 により乗算される。実効的なkクロック・サイクルが経過した後に、レジスタ Y_0 の内容は、積 $X_0 \cdot J_0$ の最下位のkビットになる。さらに、最初の実効的なkクロック・サイクルが経過した後に、ML2はリセットされる。ここで、直列入力形のマルチプレクサ M3 1; 4は、Xの流れをNの流れに切り替える。レジスタ Y_0 内のデータは、 J_0 に取って代わり、ラッチ2に並列にロードされる。さらに、ラッチ2の出力は、 $Y_0 \cdot N$ の流れに切り替えられる。次の $n+k$ クロック・サイクル期間においては、ML2からの直列の出力結果は、 $Y_0 \cdot N$ になる。実効的なkクロック・サイクル期間だけ遅延されたXは、今度は、Ad2において加算処理がなされ、ML2の積を生成する。この結果、 $Z = X + Y_0 \cdot$

Nが得られる。ここで、Zは、最下位のkビットがすべて0であるような数である。

【0172】Ad2の最初のkビットはすべて0なので、この最初のkビットは無視される。そして、次のnビットが、Sのレジスタに直列に戻される。繰り返し動作の最終的な値は、Nに等しいかまたは大きい（この場合には、この値をNから引くことが必要である）。すなわち、 $S(1) \neq S(1) \bmod N$ が成立する。SがNよりも大きいかなんかを検出するために、Sub3においては、nビット長の $(Z/2^n)$ の流れからNが減算される。しかしながら、この場合、n回目のボロービットのみが、ボロー保持用フリップフロップに記憶される。

【0173】もし、このボロービットのビット値が0であるか、または、Ad2の最終キャリービットCOのビット値が1であれば、Sのレジスタの最新の値はNよりも大きい。最初の繰り返し動作の始まりにおいては、 $S(1) \bmod N$ に対し限定された合同の関係を有する数がSのレジスタ内に存在する。 J_0 、BおよびNのレジスタは、最初にロードされたオリジナルな値を保持する。そして、データを前もって保持するための A_i のレジスタは A_1 を保持する。

【0174】第3段階：その後のB・ A_i の繰り返し動作の実行

Aの次のキャラクタである A_1 が、ラッチ1およびML1の並列入力に転送される。次のB・ A_i の繰り返し動作およびそれに続く繰り返し動作の期間中、各々の繰り返し動作の最後において、 $S(i) \bmod N$ に対し限定された合同の関係を有する数Sが存在する。もし、 $S(i)$ がNよりも大きければ、Sub2において $S(i)$ からNが引かれる。

【0175】各々の繰り返し動作が始まるときに、CPUは、Aの次のキャラクタである A_1 を、データを前もって保持するための A_i のレジスタにロードしておく。

$\rho(B \cdot B)N$ の2乗動作

通常のべき乗処理の最初の動作は、2乗動作である。この2乗動作は、Bのレジスタにロードされた乗数Aと、 A_i のレジスタにロードされた被乗数との通常の乗算と同じような手順で行われる。ただし、この場合、ビット数は、前述したようにkビット分だけ増加する。さらにその後の2乗動作は、Bのレジスタ内に存在するような限定された合同の関係を有するオペランド（乗数および被乗数）により遂行される。

【0176】上記の $\rho(B \cdot B)N$ のような2乗動作が遂行されている間、 J_0 、S、BおよびNの各レジスタは、その出力側で、前回の乗算および2乗処理により得られた値を変えずにそのままロードされる。しかしながら、この場合は、繰り返し動作において、 A_i のレジスタは、Bのレジスタ内に存在するkビットのキャラクタから派生する新しいキャラクタをロードしなければならない。

【0177】上記の連続的な2乗動作において、 A_i のレジスタは、Bのレジスタからオンザフライ式に予めロードされる。CPUが、一度、2乗処理を遂行するように指示を与えると、それ以降の2乗動作は問題なく遂行される。Bのレジスタにロードされる $B(i)'$ は、Sub 1を通して流れるBの一部分である(B_i の中で、既にNよりも小さい部分)

第1段階: $B \cdot B_0$ の繰り返し動作

まず、前回の計算処理から導き出されるような、Sに対し限定された合同の関係を有する最新の数が、Bのレジスタ内に存在するものとする。

【0178】レジスタB、Nの最下位のkビットは、周期的に右方向へのシフト動作を行う。さらに、実効的なkクロック・サイクルの後に、レジスタB、Nは、オリジナルの状態に復帰する。レジスタB内の値は、適当なBの値か、または、次のp領域での乗算を遂行するために使用される $B-N$ の値である。したがって、最初の丸めにおいて、レジスタ A_i は、 B_0 または $B-N$ の最下位のkビットでもって予めロードしなければならない。ここで、 B_0 は、レジスタB内に存在する値である。

【0179】この最初のkビットの回転動作の目的はレジスタ A_i へのプリロードの最初のkビットがSub 1を通して流れることを可能にするためである。直列にロードされた直後に、 A_i はLatch 1にアンロードされ、 A_i プリロードレジスタはBの第2の文字である B_1 をロードするために自由にされる。このおおよび引き続く操作の間、Borrow 2信号がセットあるいはリセットされた時にSub 1からの出力は正であり、常にNより小である。

【0180】全ての値がレジスタにロードされると、説明されるようにBがローテイトした時に B_1 が A_i レジスタ中にロードされる点(乗算においてCPUは A_i レジスタにロードすることを思い出すこと。)を除くと、以前に説明したようにこの最初の乗算は $B \cdot A_0$ に対して実行される。第2のkビット文字 B_1 はBストリームから発生するために、この最初の $B \cdot B_0$ 処理の間に B_1 セグメントは、次の開平演算、すなわち $B \cdot B_1$ 繰り返しのために直列的に A_i プリロードレジスタ中にオンザフライ処理で切り換えられる。

第2段階: $B \cdot B_1$ 繰り返し動作

A_i のレジスタ中にロードされた値 B_1 は出力ラッチLatch 1に転送される。次の $n+2k$ (すなわち $n+64$)クロックサイクル中に $B \cdot B_1$ に対する乗算処理が上述のように実行される。

【0181】前回と同様Borrow 1 およびBorrow 2 信号

は、BおよびSレジスタから発生する流れからNが減算されうるか否かを決定する。もしSレジスタ中の値がNより大であるかあるいは等しければBorrow 1はセットされ、減算器Sub 1においてNはSから減算される。もし必要であればm繰り返し乗算ループ完了の間にNはSから減算される。このような状況は先行する乗算あるいは開平演算の終わりにBorrow 2で検知される。

【0182】フリップフロップBorrow 1 およびBorrow 2はSub 3からの条件付きのボロウ出力の最終値を記憶している。Borrow 1は各Sの繰り返しの後にセットあるいはリセットされる。Borrow 2はBが $S(m)$ にロードされる最後の $S(m)$ 繰り返しの後にセットあるいはリセットされる。この条件付きのボロウ出力は $S(i)$ がNより大であるか否かを示す信号である。

【0183】 $B \cdot B_1$ 処理の間、文字 B_2 が減算器Sub 1中に存在する場合に文字 B_2 はオンザフライ処理で A_i プリロードレジスタ中にロードされる。

第3段階: 次の $B \cdot B_1$ 乗算繰り返し

文字 B_1 が減算器Sub 1中に存在する時には文字 B_1 の値が A_i レジスタにロードされる間に残りの $m-2$ 回の繰り返しが次のループの準備のために実行される。

【0184】限定された一致の最終結果はSおよびBレジスタ中に存在する。このデータはDOを通して直列に出力されるために、もし必要であればSub 1で修正されるであろう。

乗算ブロックの操作-Hパラメータの演算

Hを演算するために、マシンは図9に示されるようにレジスタSおよびNを使用するために再構成される。上記で既に使用した数値例を用いて演算子の操作を説明する。この構成はHの演算を $n+1$ 回で実行する。各回の実行においてSおよびNは共にローテイトされ、各ローテイトはnクロックである。各実行回においてNは回転し変化せずに帰還する。i回目の実行において、Sおよび次の減算(Next Subtract)信号は $S(i)$ の限定された一致の同等値を含んでいる。

初期条件-第1回実行

第1回目の実行の最初で、NはNレジスタ中にロードされ、第1回目の試行減算が成功したことを表すボロー検出フラグはリセットされ、Sub 1の出力フリップフロップはゼロにリセットされる。第1回目の実行中、試行除算のn番目のMSビットは“1”である。このビットは次の減算用フリップフロップ(S中にスペースはない。)を推論して記憶される。次の減算は、第1回目の実行において $S-N$ 減算を命令する。上述の $n=4$ ビットの数値例を使用して例証する。

H計算モード-初期条件

ポロー検出器の次減算フラグに格納される

まず、被除数のMSビットが
"1"であることを知る。

$N = 1011$, $n = 4$

図7参照

それ故、ポローが存在し得ない
ことを知るので、次の減算フラグを
零にリセットする。

S (0) Sレジスタの内容

----- (0) 0000 {0000} ← "仮想零"

ポロー検出次減算

信号は零である—それ故、最初の
ラウンドでM2_1:3はNをSub1
に与える—差はリーディング零を
伴うS-Nか、ちょうど2・(S-N)
となろう。

↑↑↑↑
[これら"仮想"のLS零は
試用減算によって影響されな
い。各ラウンドで、"仮想零
カウンタ"に1-0が存在す
るだろう。]

最初のクロックサイクルで、リセット
Sub1出力フリップフロップからの
零は、SからのLSビットがSub1
に与えられるとき、SのMSセルに
与えられる。

(SのLSビットは常に"仮想"LS
零カウンタから"引かれる"零で
である。)

最初のn-1クロックサイクルの間、
DiffのLSのn-1ビットが
Sに与えられる。

NはそのMSビットセルに回転バックされる。

BO (ポローアウト) シリアルストリームは

$Diff \bmod 2^n - N$ ストリーム
から結果するポローの一続きに等しい。しかし、
最後のポローのみサンプルされ、関連する
かもしれない。

第nの実効クロックサイクルにて、
DiffのMSビットが"1"であるか又は

BO="0"ならば、"次減算"が次の
ラウンドの減算用にフラグを上げる。

最初のラウンドにて 2^n からNがひかれ、2 (LS零挿入) を乗じられた結果の
nビットはSレジスタにリターンされる。ただし、"推論によって"ポロー検出
次減算レジスタに格納されるMSビットを除く。

最初のラウンドの終わりにて回転する:

S (1) = 1010、次の減算=1 (BO=1)、そして、次のラウンドでSu
b1にはS-Nの減算が存在しない。

Hパラメータ計算-第2ラウンドポロ-検出の次の減算フラグに格納される

まず、第2ラウンドの減算が、
Sub 2にて”検出された”
BO=”1”として成功しないだろう
ことを知る。

N=1011, n=4

S (1) 最初のラウンド後のSレ
ジスタの内容

----- (1) 1010 {000} ←”3仮想
零”が残る

ポロ-検出次減算

信号は1-それ故このラウンドで
M2_1; 3はSub 1に零を
与えよう-Diff=2・S

減算が存在しなかった

(SのLSビットは再び”仮想”
LS零カウンタ”から”引かれた”
零である。)

続くn-1クロックサイクルの間

Diff=2・SのLSのn-1

ビットがSレジスタに与えられる。

NはそのMSビットセルに回転バックされる。

DiffのMSビットが”1”のとき、

次のラウンドでS-Nを減算しなければ

ならないことを知る。

サンプルされたBOは不適切である。

Diff=1 0100かつS(2)=0100 そして、次のラウンドでSu

b1にてS-Nの減算があることを知る。

[0186]

Hパラメータ計算-第3ラウンドボロー検出器の次減算フラグに格納される

まず、DiffのMSビットが
"1"だったので、第3ラウンド
の減算が成功するだろうことを
知る。

N=1011, n=4

S(2) 第2ラウンド後のSレジ
スタの内容

----- (0) 0100 {00} ← "2仮想零"
残る

ボロー検出次減算

信号は零。DiffからNが
引かれる。

続くn-1クロックサイクルの間、
Diff=2(S-N)のLSの
n-1ビットがSレジスタに逆供給
される。

DiffのMSビットが次の
ラウンドのSub1において
"1"なので、S-Nの

減算をしなければならない。

Diff=1 0010 かつ S(3)=0010、次の減算=0 かつ
次のラウンドでSub1においてS-Nの減算があることを知る。

【0187】

Hパラメータ計算-第4ラウンドボロー検出器の次減算フラグに格納される

まず、DiffのMSビットが"1"
だったので、第4ラウンド減算が
成功するだろうことを知る。

N=1011, n=4

S(3) 第3ラウンド後の
Sレジスタの内容

----- (0) 0010 {0} ← "1仮想零"
残る

ボロー検出次減算

信号は零。NがDiffから
引かれる。

ボローBO="0"がなかったので、
次のラウンドでS-Nの減算をする。

Diff=0 1110 かつ S(4)=1110、次減算=0、そして、
次のラウンドではSub1においてS-Nの減算があろうことを知る。

【0188】

Hパラメータ計算-第n+1(第5)ラウンド

ボロー検出器の次減算フラグに格納される

まず、DiffのMSビットが"1"だったので、第4ラウンド減算が成功するだろうことを知る。

N=1011, n=4

S(4) 第4ラウンド後の
Sレジスタの内容

----- (0) 1110 {} ← "ノー仮想" が
残る
最終ラウンド

ボロー検出次減算

信号は零。NがDiffから引かれる。

Diff=0 0011 かつ S(5)=0011, が残りである。それはHの値である。

【図面の簡単な説明】

【図1】本発明の一実施例の装置構成を示すブロック図 20 である。

【図2】本発明の一実施例のモジュール・乗算回路を示すブロック図である。

【図3】本発明の一実施例の特殊なモジュール・乗算回路を示すブロック図である。

【図4】本発明の一実施例による繰り返し動作と乗算動作との間の時間的な関係を示す図である。

【図5】直列／並列乗算器のセルの構成を示す回路図である。

【図6】8ビットの直列／並列乗算器の構成を示す回路 30

図である。

【図7】直列加算器の構成を示す回路図である。

【図8】直列減算器の構成を示す回路図である。

【図9】Hパラメータを計算するためのアーキテクチャを示すブロック図である。

【符号の説明】

10～12…レジスタ

13～15…マルチプレクサ

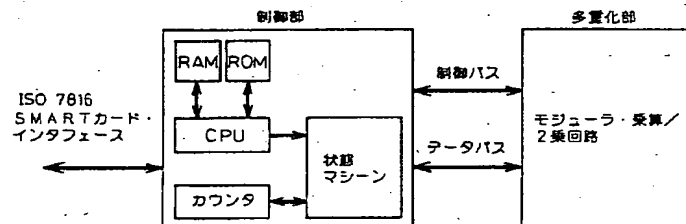
16～18…レジスタ

27～31…加算／減算手段

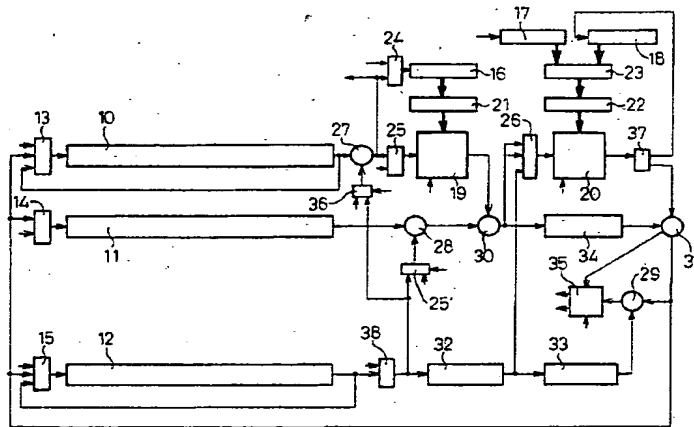
32～34…ディレイ手段

35…ボロー検出器

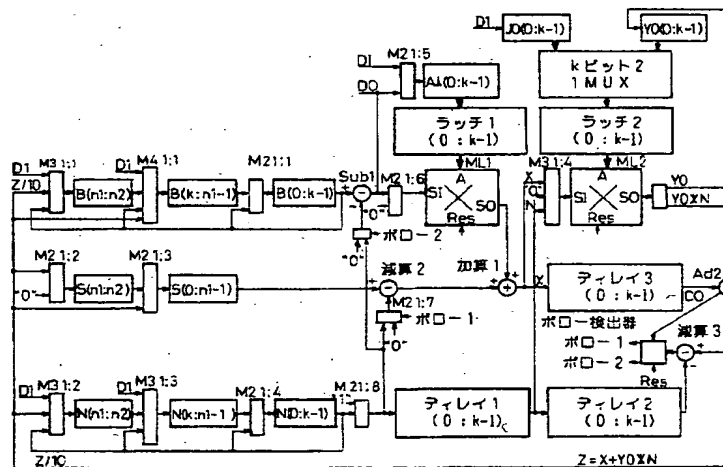
【図1】



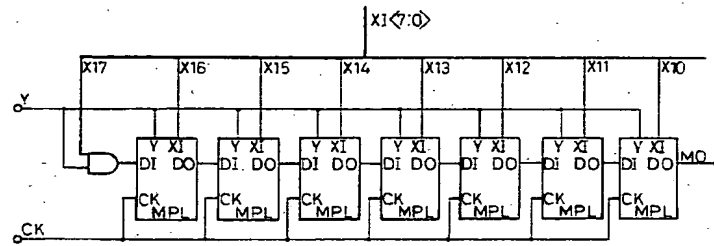
【図2】



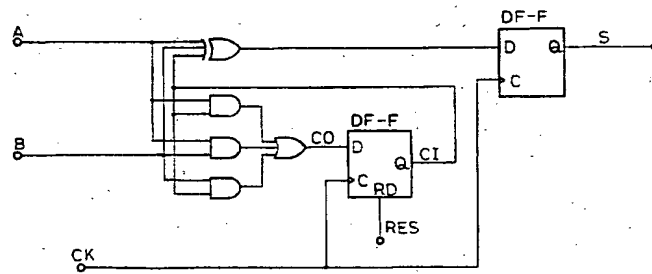
【図3】



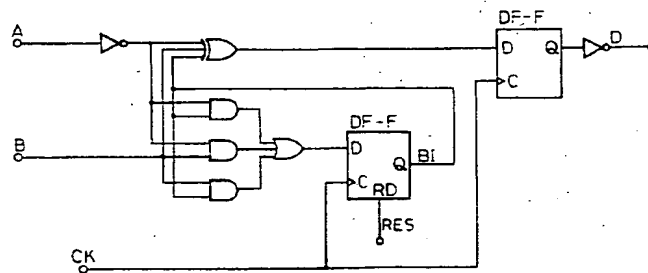
【図6】



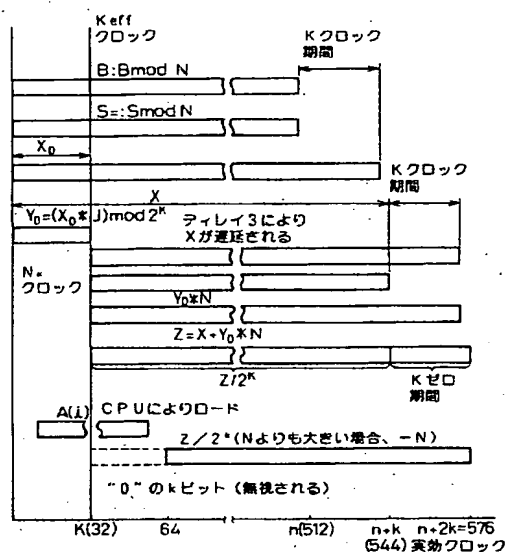
【図7】



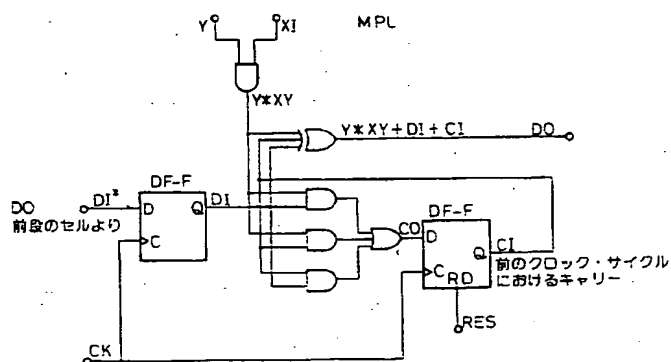
【図8】



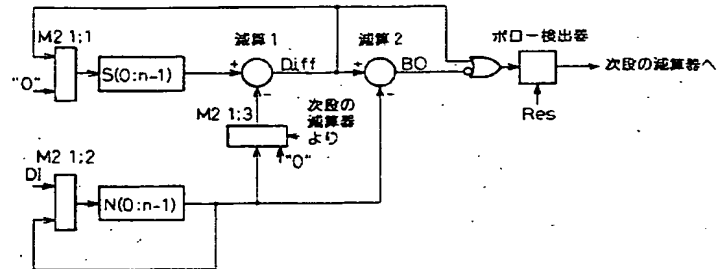
【圖4】



【图 5】



【図9】



フロントページの続き

(72)発明者 イタイ ドローア
イスラエル国, ビアーシェバ, ミブツア
ナチヨン 76/32

(72)発明者 イザーク ハダッド
イスラエル国, ビアーシェバ 84434, デ
レチ ハシャロム 105/3

(72)発明者 ベンジャミン アラジ
イスラエル国, オマー 84965, シガロン
ストリート 38